

# UCSS



UNIVERSIDAD CATÓLICA SEÑOR DE LA DIVINIDAD

---

**INTEGRACIÓN CONTINUA DE CODIGO ABIERTO DE GITHUB CON GENKINS  
PARA UN PROYECTO INTRANET EN FLUTTER**

---

**FACULTAD DE INGENIERÍA  
CARRERA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**Asignatura:**

FUNDAMENTO DE SISTEMAS OPERATIVOS

**Docente:**

Vásquez Reyes Eduardo

**Estudiante:**

Villalobos Silva Kleiwin

**CICLO ACADÉMICO**

**2024 - 2**

## Índice del proyecto

<b>Sección 0: Desarrollo de la aplicación .....</b>	<b>4</b>
<b>INTRODUCCION .....</b>	<b>4</b>
<b>DESARROLLO .....</b>	<b>4</b>
Requerimientos Funcionales:.....	4
Requerimientos No Funcionales:.....	5
Consideraciones Adicionales: .....	5
MODELO FÍSICO DEL BD.....	6
SCRIP INICIAL DEL BD .....	6
<b>SERVICIO NOTAS ServidorNotas .....</b>	<b>17</b>
fotos .....	17
Conexion1.php .....	18
Negocio1.php.....	19
Controla1.php.....	27
<b>FLUTTER.....</b>	<b>31</b>
Clases .....	31
Credenciales.dart .....	32
Login.....	37
PAGINAS DE NAVEGACIÓN .....	50
Visor app.....	96
<b>Sección 1: Procedimiento documentado de la instalación de Jenkins: .....</b>	<b>121</b>
1. Requisitos previos .....	121
2. Pasos de instalaciones .....	122
2.1 Descargar Jenkins.....	122
2.2. Instalar Jenkins: .....	123
2.3. Verificar la instalación de Jenkins:.....	128
2.4. Desbloquear Jenkins .....	129
2.5. Configurar usuario administrador .....	131
2.6. Instalar plugins recomendados .....	132
2.8. Descargar NGROK: .....	137
2.9. Instalación del GITHUB.....	140
2.10. Instalar Visual Studio Code:.....	143
<b>Sección 2:Procedimiento de configuración de Jenkins para que realice la sincronización entre un proyecto en GitHub y el repositorio en el cliente (PC): .....</b>	<b>146</b>
2.1. Creación del repositorio en GitHub:.....	146
2.2. Creación del WebHook: .....	155
2.3. Creación del Job (Jenkins): .....	157
<b>Sección 3: Ejecución de la sincronización: .....</b>	<b>165</b>
3.1. Para ello debemos repetir esto: .....	165

<b>3.2. Gestión cambios .....</b>	<b>169</b>
<b>Sección 4: Gestión del repositorio para el proyecto.....</b>	<b>170</b>
<b>Paso 1: Crear una Cuenta en GitHub.....</b>	<b>170</b>
<b>Paso 2: Crear un Repositorio en GitHub .....</b>	<b>170</b>
<b>Paso 3: Subir Tu Proyecto a GitHub.....</b>	<b>173</b>
3.1. Preparar tu Proyecto Local .....	173
3.2. Inicializar un Repositorio Git Local .....	173
3.3. Configurar el Repositorio Remoto.....	174
3.4. Agregar los Archivos y Realizar un Commit.....	174
3.5. Subir los Archivos a GitHub .....	176
<b>Paso 4: Configurar una Licencia de Código Abierto (Opcional).....</b>	<b>177</b>
<b>Paso 5: Agregar un Archivo de Contribución .....</b>	<b>179</b>

## **Sección 0: Desarrollo de la aplicación**

### **Desarrollo de aplicación en flutter para administrar datos de profesores y estudiantes, (Perú, 2024-II).**

#### **INTRODUCCION**

En el contexto educativo moderno, la gestión eficiente de datos es crucial para asegurar el funcionamiento efectivo de instituciones académicas. En respuesta a esta necesidad, el desarrollo de aplicaciones se ha convertido en una herramienta poderosa para facilitar la administración de información tanto de profesores como de estudiantes. En el año académico 2024-I en Perú, se presenta un escenario dinámico donde la implementación de tecnologías innovadoras como Flutter ofrece soluciones ágiles y accesibles para gestionar datos de manera centralizada y efectiva.

#### **DESARROLLO**

Para desarrollar una aplicación en Flutter que administre datos de profesores y estudiantes en el contexto académico peruano durante el período 2024-I, es importante definir los requerimientos funcionales y no funcionales que guiarán el desarrollo del proyecto. Aquí te presento una lista básica de requerimientos que podrías considerar:

##### **Requerimientos Funcionales:**

###### **1. Autenticación y Seguridad:**

- Los usuarios (administradores, profesores, estudiantes) deben poder autenticarse de manera segura utilizando credenciales únicas.
- Implementación de roles de usuario para controlar el acceso a diferentes funciones y datos.

###### **2. Gestión de Profesores:**

- Registrar y mantener información detallada de cada profesor (código, nombre, correo electrónico, especialidad).
- Permitir mostrar información de perfiles de profesores por parte de los administradores que corresponde a ellos.

###### **3. Gestión de Estudiantes:**

- Mantener registros de estudiantes (notas).
- Estudiantes en diferentes cursos .

###### **4. Gestión de Cursos:**

- Administrar la lista de cursos ofrecidos por la institución (código, docente, nombre, créditos, costo).
- Administrar todos los estudiantes en ese curso (solo calificaciones por parte del profesor)
- El estudiante solo puede visualizar notas en los cursos inscritos.

- El profesor puede ingresar evaluaciones o modificar las mismas.

##### **5. Calificaciones y Evaluaciones:**

- Permitir a los profesores ingresar calificaciones y evaluaciones de manera segura (parcial1, parcial2, entre otros).
- Mantener un registro histórico de las calificaciones de los estudiantes.
- Generar informes de desempeño académico para estudiantes (Boletas de notas por curso y por estudiante en diferentes cursos).
- Consultar plan académico o de estudio por ciclo del estudiante.

#### Requerimientos No Funcionales:

##### **1. Usabilidad y Experiencia de Usuario:**

- Interfaz de usuario intuitiva y amigable para facilitar la navegación y el uso por parte de usuarios no técnicos.
- Tiempos de carga rápidos y respuesta ágil de la aplicación.

##### **2. Seguridad de Datos:**

- Implementación de medidas robustas de seguridad para proteger la integridad y confidencialidad de los datos sensibles.
- Cumplimiento con normativas de protección de datos personales (GDPR, Ley de Protección de Datos, etc.).

##### **3. Compatibilidad y Escalabilidad:**

- Soporte para múltiples plataformas (Android, iOS) utilizando Flutter.
- Diseño modular y escalable que permita la adición de nuevas funcionalidades en el futuro.

##### **4. Mantenimiento y Actualizaciones:**

- Facilidad para realizar mantenimiento y aplicar actualizaciones de forma regular.
- Documentación clara y detallada del código y funcionalidades para futuras revisiones y extensiones.

##### **5. Integración con Sistemas Existentes:**

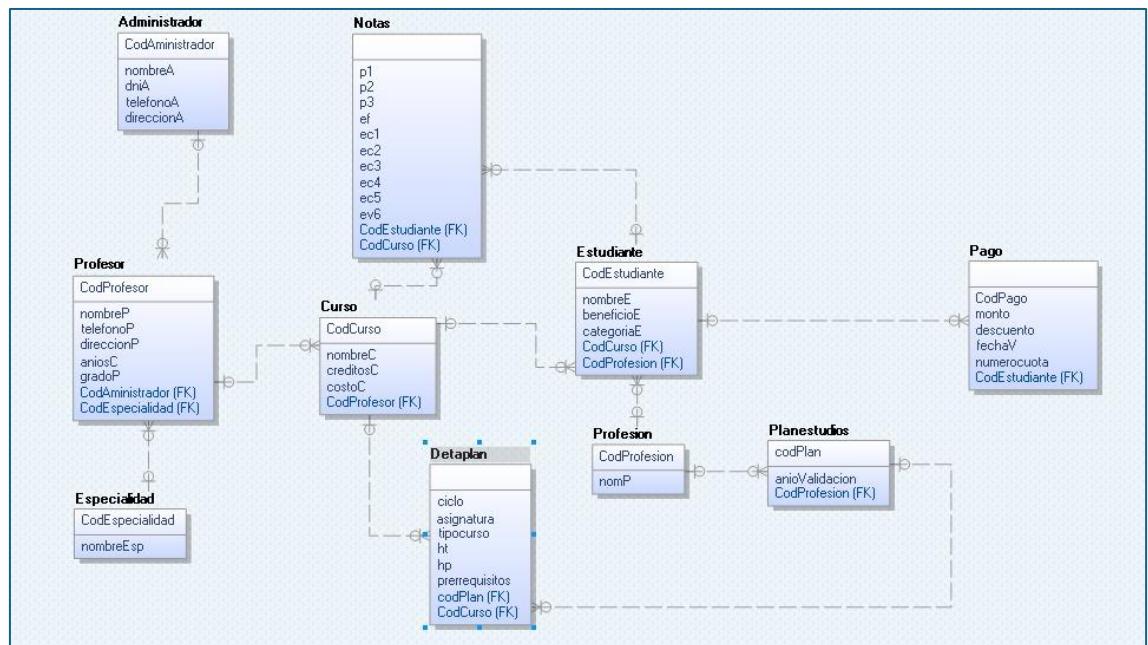
- Capacidad para integrarse con sistemas de gestión académica existentes (si es aplicable).

#### Consideraciones Adicionales:

- **Interfaz Gráfica:** Diseño adaptable y estéticamente agradable que refleje la identidad visual de la institución educativa.
- **Soporte Técnico:** Disponibilidad de soporte técnico y capacitación para usuarios finales.
- **Pruebas y Validación:** Implementación de pruebas exhaustivas (unitarias, integración, aceptación) para asegurar la calidad del software.

Estos requerimientos proporcionan una base sólida para el desarrollo de una aplicación efectiva para administrar datos académicos en el contexto educativo peruano durante el período 2024-I. Adaptar estos requerimientos específicamente a las necesidades y políticas de la institución educativa será crucial para el éxito del proyecto.

## MODELO FÍSICO DEL BD.



## SCRIP INICIAL DEL BD

```
create or replace database bdfinalpc3;
use bdfinalpc3;
```

```
CREATE TABLE Administrador
(
    CodAdministrador  VARCHAR(20) NOT NULL primary key,
    nombreA          VARCHAR(40),
    dniA              VARCHAR(8),
    telefonoA        VARCHAR(10),
    direccionA       VARCHAR(50)
);
```

```
CREATE TABLE CursoProfesor
```

```
(  
    CodCurso      INT (6) not null primary key,  
    CodProfesor   VARCHAR(20),  
    nombreC       VARCHAR(60) ,  
    creditosC    VARCHAR(3) ,  
    costoC       NUMERIC(8,2)  
  
);
```

CREATE or replace TABLE Detaplan

```
(  
    ciclo        VARCHAR(3) ,  
    codPlan     VARCHAR(20),  
    codigo       int (6),  
    asignatura   varchar(60),  
    tipocurso    VARCHAR(20) ,  
    ht          VARCHAR(20) ,  
    hp          VARCHAR(20) ,  
    prerequisitos  VARCHAR(40)  
  
);
```

CREATE TABLE Especialidad

```
(  
    CodEspecialidad  VARCHAR(20) NOT NULL primary key,  
    nombreEsp        VARCHAR(40)  
  
);
```

CREATE TABLE Estudiante

```
(  
    CodEstudiante   INT (10),  
    CodCurso        INT(6) ,  
    CodProfesion   VARCHAR(20),  
    nombreE         VARCHAR(40) ,  
    beneficioE     VARCHAR(20) ,  
    categoriaE     VARCHAR(3)  
  
);
```

CREATE TABLE Notas

```
(  
    CodEstudiante   INT (10),  
    CodCurso        INT (6),  
    p1            FLOAT ,  
    p2            FLOAT ,
```

```
    p3      FLOAT ,  
    ef      FLOAT ,  
    ec1     FLOAT ,  
    ec2     FLOAT ,  
    ec3     FLOAT ,  
    ec4     FLOAT ,  
    ec5     FLOAT ,  
    ev6     FLOAT  
  
);  
  
CREATE TABLE Pago  
(  
    CodPago      VARCHAR(20),  
    CodEstudiante INT(10) ,  
    fechaV       datetime ,  
    numerocuota  VARCHAR(20) ,  
    monto        DECIMAL(20,8) ,  
    descuento    VARCHAR(20)  
);
```

```
CREATE TABLE Planestudios  
(  
    codPlan      VARCHAR(20) not null primary key,  
    CodProfesion VARCHAR(20) ,  
    anioValidacion VARCHAR(20)  
  
);
```

```
CREATE TABLE Profesion  
(  
    CodProfesion  VARCHAR(20) not null primary key,  
    nomP         VARCHAR(40)  
);
```

```
CREATE TABLE Profesor  
(  
    CodProfesor    VARCHAR(20) not null primary key,  
    CodAdministrador VARCHAR(20) ,  
    CodEspecialidad VARCHAR(20) ,  
    nombreP        VARCHAR(60) ,  
    telefonoP      VARCHAR(10) ,  
    direccionP    VARCHAR(50) ,  
    gradoP         VARCHAR(20) ,  
    aniosC         VARCHAR(20)
```

);

-- Inserts para la tabla Administrador

```
INSERT INTO Administrador (CodAdministrador, nombreA, dniA, telefonoA,  
direccionA)  
VALUES  
('ADM001', 'Juan Pérez', '12345678', '999888777', 'Av. Principal 123'),  
('ADM002', 'María González', '87654321', '666555444', 'Calle Secundaria 456'),  
('ADM003', 'Julio Piedra', '87654321', '666555444', 'Calle Secundaria 456'),  
('ADM004', 'José Silva', '87654321', '666555444', 'Calle Secundaria 456')  
;
```

-- Inserts para la tabla Especialidad

```
INSERT INTO Especialidad (CodEspecialidad, nombreEsp)  
VALUES  
('ESP001', 'Ingeniería de Software'),  
('ESP002', 'Bases de Datos y Programación'),  
('ESP003', 'Programación'),  
('ESP004', 'Algorítmica'),  
('ESP005', 'Ingeniería Sistemas'),  
('ESP006', 'Estadística')  
;
```

-- Inserts para la tabla Profesion

```
INSERT INTO Profesion (CodProfesion, nomP)  
VALUES  
('PROF001', 'Ingeniería de Sistemas'),  
('PROF002', 'Ingeniería Industrial'),  
('PROF003', 'Ingeniería Ambiental')  
;
```

-- Inserts para la tabla Profesor

```
INSERT INTO Profesor (CodProfesor, CodAdministrador, CodEspecialidad, nombreP,  
telefonoP, direccionP, gradoP, aniosC)  
VALUES  
('PROF001', 'ADM001', 'ESP001', 'Luis Ramírez', '977666555', 'Av. Principal 456',  
'Maestría', '10'),  
('PROF002', 'ADM002', 'ESP002', 'Ana López', '955444333', 'Calle Secundaria 789',  
'Los Olivos', 'Doctorado', '6'),
```

```
('PROF003', 'ADM001', 'ESP003', 'Ramón Quispe', '977666555', 'Av. Principal 476,  
SMP', 'Maestría', '4'),  
(('PROF004', 'ADM002', 'ESP004', 'Adrian Piedra', '955444333', 'Calle Secundaria 79,  
Lima', 'Doctorado', '5'),  
(('PROF005', 'ADM001', 'ESP005', 'Carlos Cardenas', '977666555', 'Av. Principal 87,  
Los Olivos', 'Maestría', '8'),  
(('PROF006', 'ADM002', 'ESP006', 'José Huaman', '955444333', 'Calle Secundaria 78,  
SMP', 'Doctorado', '9')  
;
```

-- Inserts para la tabla Curso

```
INSERT INTO CursoProfesor (CodCurso, CodProfesor, nombreC, creditosC,  
costoC)  
VALUES  
(2000, 'PROF002', 'Programación Computadoras 3', '3', 517.00),  
(2001, 'PROF002', 'Bases de Datos 2', '3', 517.00),  
(2002, 'PROF004', 'Microprocesadores', '3', 517.00),  
(2003, 'PROF001', 'Ingeniería de Sofware 1', '3', 517.00),  
(2004, 'PROF005', 'Sistemas Digitales', '3', 517.00),  
(2005, 'PROF006', 'Investigación operativa', '4', 670.00)  
;
```

-- Inserts para la tabla Detaplan

```
INSERT INTO Detaplan (ciclo, codPlan, codigo, asignatura, tipocurso, ht, hp,  
prerrequisitos)  
VALUES  
('1', 'PLAN003', 150286, 'ACTIVIDADES I', 'O', '0', '2', '1'),  
('1', 'PLAN003', 170001, 'ANTROPOLOGIA RELIGIOSA', 'O', '4', '0', '4'),  
('1', 'PLAN003', 200258, 'ESTRUCTURAS DISCRETAS I', 'O', '2', '2', '3'),  
('1', 'PLAN003', 200257, 'FUNDAMENTOS DE SISTEMAS DE INFORMACION Y  
CIENCIAS DE LA COMPUTACION', 'O', '2', '4', '4'),  
('1', 'PLAN003', 150285, 'INGLES I', 'O', '0', '2', '1'),  
('1', 'PLAN003', 150283, 'LENGUA: COMUNICACION ESCRITA ACADEMICA', 'O', '4',  
'2', '5'),  
('1', 'PLAN003', 131042, 'MATEMATICA BASICA PARA INGENIERIA', 'O', '2', '4', '4'),  
('1', 'PLAN003', 132095, 'HERRAMIENTAS DE OFIMATICA', 'E', '2', '2', '3'),  
('2', 'PLAN003', 200260, 'ALGORITMOS Y ESTRUCTURAS DE DATOS', 'O', '2', '4', '(I)  
200257 - (I) 200258'),  
('2', 'PLAN003', 131043, 'CALCULO DIFERENCIAL', 'O', '2', '4', '(I) 131042'),  
('2', 'PLAN003', 132104, 'FISICA 1', 'O', '2', '2', '(I) 131042'),  
('2', 'PLAN003', 150288, 'INGLES II', 'O', '0', '2', '(I) 150285'),  
('2', 'PLAN003', 150284, 'REDACCION ACADEMICA', 'O', '4', '2', '(I) 150283'),  
('2', 'PLAN003', 170009, 'TEOLOGIA I', 'O', '4', '0', '(I) 170001'),
```

('2', 'PLAN003', 200291, 'TEORIA Y DINAMICA DE SISTEMAS', 'O', '2', '2', '(I) 200257'),  
('3', 'PLAN003', 132105, 'FISICA 2', 'O', '2', '2', '(II) 132104'),  
('3', 'PLAN003', 200292, 'GESTION DE PROCESOS DE NEGOCIOS', 'O', '2', '2', '(II)  
200291'),  
('3', 'PLAN003', 200310, 'INGENIERIA ECONOMICA FINANCIERA', 'O', '2', '2', '(II)  
131043'),  
('3', 'PLAN003', 150289, 'INGLES III', 'O', '0', '2', '(II) 150288'),  
('3', 'PLAN003', 200263, 'INNOVACION Y DESARROLLO TECNOLOGICO', 'O', '2', '2',  
'(II) 200260'),  
('3', 'PLAN003', 200293, 'PROGRAMACION DE COMPUTADORAS I', 'O', '2', '4', '(II)  
200260'),  
('3', 'PLAN003', 200261, 'PROGRAMACION ORIENTADA A OBJETOS', 'O', '2', '2', '(II)  
200260'),  
('3', 'PLAN003', 170010, 'TEOLOGIA II', 'O', '4', '0', '(II) 170009'),  
('3', 'PLAN003', 200307, 'TALLER DE CREATIVIDAD TECNOLOGICA', 'E', '2', '2', '(I)  
200257'),  
('4', 'PLAN003', 131016, 'ASERTIVIDAD Y COMUNICACION EFECTIVA', 'O', '0', '2', '(II)  
150284 - (III) 170010'),  
('4', 'PLAN003', 200223, 'BASE DE DATOS 1', 'O', '2', '2', '(III) 200261'),  
('4', 'PLAN003', 200196, 'DISEÑO Y EVALUACION DE PROYECTOS', 'O', '2', '2', '(III)  
200310'),  
('4', 'PLAN003', 200264, 'ESTADISTICA Y PROBABILIDADES', 'O', '2', '2', '(III) 200310'),  
('4', 'PLAN003', 150290, 'INGLES IV', 'O', '0', '2', '(III) 150289'),  
('4', 'PLAN003', 200296, 'PROCESOS DE SOFTWARE', 'O', '2', '2', '(III) 200292 - (III)  
200263'),  
('4', 'PLAN003', 200294, 'PROGRAMACION DE COMPUTADORAS II', 'O', '2', '4', '(III)  
200293'),  
('4', 'PLAN003', 200225, 'SISTEMAS DIGITALES', 'O', '2', '2', '(III) 132105'),  
('4', 'PLAN003', 200265, 'DESARROLLO BASADO EN PLATAFORMAS', 'E', '2', '2', '(III)  
200261'),  
('5', 'PLAN003', 200242, 'BASE DE DATOS 2', 'O', '2', '2', '(IV) 200223'),  
('5', 'PLAN003', 200268, 'FUNDAMENTOS DE SISTEMAS OPERATIVOS', 'O', '2', '2',  
'(IV) 200225'),  
('5', 'PLAN003', 200224, 'INGENIERIA DE SOFTWARE 1', 'O', '2', '2', '(IV) 200296 - (IV)  
131016'),  
('5', 'PLAN003', 150291, 'INGLES V', 'O', '0', '2', '(IV) 150290'),  
('5', 'PLAN003', 131023, 'INVESTIGACION OPERATIVA', 'O', '2', '4', '(IV) 200264 - (IV)  
200196'),  
('5', 'PLAN003', 200267, 'MICROPROCESADORES Y ARQUITECTURA DE  
COMPUTADORAS', 'O', '2', '2', '(IV) 200225'),  
('5', 'PLAN003', 200295, 'PROGRAMACION DE COMPUTADORAS III', 'O', '2', '2', '(IV)  
200294'),  
('5', 'PLAN003', 200266, 'DISEÑO DE ALGORITMOS', 'E', '2', '2', '(IV) 200294'),  
('6', 'PLAN003', 200297, 'EXPERIENCIA DE USUARIO Y USABILIDAD DEL  
SOFTWARE', 'O', '2', '2', '(V) 200224'),  
('6', 'PLAN003', 200231, 'INGENIERIA DE SOFTWARE 2', 'O', '2', '2', '(V) 200224'),  
('6', 'PLAN003', 150294, 'INGLES VI', 'O', '0', '2', '(V) 150291'),

('6', 'PLAN003', 200232, 'INTELIGENCIA DE NEGOCIOS Y ANALISIS DE DATOS', 'O', '2', '2', '(V) 200242'),  
('6', 'PLAN003', 200270, 'INTERACCION HUMANO COMPUTADOR', 'O', '2', '2', '(V) 200268 - (V) 200267'),  
('6', 'PLAN003', 200298, 'MARKETING DIGITAL', 'O', '2', '2', '(V) 200242'),  
('6', 'PLAN003', 200271, 'PROCESOS ESTOCASTICOS', 'O', '2', '2', '(V) 131023'),  
('6', 'PLAN003', 200269, 'REDES Y COMUNICACIONES', 'O', '2', '2', '(V) 200268'),  
('6', 'PLAN003', 200272, 'MODELOS Y SIMULACION', 'E', '2', '2', '(V) 131023'),  
('6', 'PLAN003', 200156, 'TECNOLOGIAS EMERGENTES I', 'E', '2', '2', '(V) 200268'),  
('7', 'PLAN003', 200275, 'BIG DATA', 'O', '2', '4', '(VI) 200271'),  
('7', 'PLAN003', 200300, 'CALIDAD DE SOFTWARE', 'O', '2', '2', '(VI) 200297'),  
('7', 'PLAN003', 200299, 'GESTION DE PROYECTOS DE TI', 'O', '2', '2', '(VI) 200231'),  
('7', 'PLAN003', 200276, 'INFRAESTRUCTURA DE TECNOLOGIAS DE INFORMACION', 'O', '2', '2', '(VI) 200269 - (VI) 200270'),  
('7', 'PLAN003', 150297, 'INGLES VII', 'O', '0', '2', '(VI) 150294'),  
('7', 'PLAN003', 200301, 'INNOVACION DE SISTEMAS DE INFORMACION Y NUEVAS TECNOLOGIAS', 'O', '2', '2', '(VI) 200232 - (VI) 200298'),  
('7', 'PLAN003', 100077, 'LIDERAZGO EMPRESARIAL', 'O', '2', '2', '(VI) 200232'),  
('7', 'PLAN003', 200311, 'SI DE GESTION EMPRESARIAL', 'O', '2', '2', '(VI) 200231'),  
('7', 'PLAN003', 200243, 'LENGUAJE Y COMPILADORES', 'E', '2', '2', '(V) 131023'),  
('7', 'PLAN003', 200256, 'TOPICOS AVANZADOS DE DISEÑO Y GESTION DE REDES', 'E', '2', '2', '(VI) 200269'),  
('8', 'PLAN003', 200280, 'COMPUTACION EN LA NUBE', 'O', '2', '2', '(VII) 200275'),  
('8', 'PLAN003', 200281, 'DISEÑO Y GESTION DE ARQUITECTURA EMPRESARIAL', 'O', '2', '2', '(VII) 200311 - (VII) 200276'),  
('8', 'PLAN003', 200302, 'GESTION DE RIESGOS Y SEGURIDAD DE TI', 'O', '2', '4', '(VII) 200311'),  
('8', 'PLAN003', 200279, 'IMPLEMENTACION Y DESPLIEGUE DE SOLUCIONES INFORMATICAS', 'O', '2', '4', '(VII) 200311 - (VII) 200300'),  
('8', 'PLAN003', 150298, 'INGLES VIII', 'O', '0', '2', '(VII) 150297'),  
('8', 'PLAN003', 250010, 'METODOLOGIA DE LA INVESTIGACION', 'O', '2', '2', '(VII) 200299 - (VII) 100077'),  
('8', 'PLAN003', 200278, 'SEGURIDAD EN COMPUTADORAS', 'O', '2', '2', '(VII) 200276 - (VII) 200301'),  
('8', 'PLAN003', 200153, 'COMPUTACION GRAFICA', 'E', '2', '2', '(V) 200266'),  
('8', 'PLAN003', 200277, 'FUNDAMENTOS DE INTELIGENCIA ARTIFICIAL', 'E', '2', '4', '(V) 200266'),  
('9', 'PLAN003', 200234, 'AUDITORIA DE SISTEMAS DE INFORMACION', 'O', '2', '2', '(VIII) 200278'),  
('9', 'PLAN003', 200285, 'ETICA EN LA PROFESION Y GESTION DE PATENTES', 'O', '2', '2', '(VIII) 250010'),  
('9', 'PLAN003', 200284, 'FUNDAMENTOS DE CIBERSEGURIDAD', 'O', '2', '2', '(VIII) 200278 - (VIII) 200280'),  
('9', 'PLAN003', 200303, 'GESTION DE LA CONFIGURACION DEL SOFTWARE', 'O', '2', '2', '(VIII) 200279 - (VIII) 200281'),  
('9', 'PLAN003', 150300, 'INGLES IX', 'O', '0', '2', '(VIII) 150298'),

('9', 'PLAN003', 200238, 'PRACTICAS PRE PROFESIONALES DE INGENIERIA DE SISTEMAS', 'O', '0', '2', '(VIII) 200279'),  
('9', 'PLAN003', 200237, 'SEMINARIO DE INVESTIGACION 1 INGENIERIA DE SISTEMAS', 'O', '2', '4', '(VIII) 250010'),  
('9', 'PLAN003', 200309, 'CONFIGURACION DE DISPOSITIVOS DE RED', 'E', '2', '2', '(VIII) 200279'),  
('9', 'PLAN003', 200308, 'GESTION DE SERVICIOS DE TI', 'E', '2', '2', '(VIII) 200279'),  
('9', 'PLAN003', 200283, 'INTERNET DE LAS COSAS', 'E', '2', '4', '(VI) 200269'),  
('9', 'PLAN003', 200282, 'ROBOTICA', 'E', '2', '2', '(VIII) 200277'),  
('10', 'PLAN003', 200290, 'EMPRENDIMIENTO Y TRANSFERENCIA TECNOLOGICA', 'O', '2', '0', '(IX) 200285 - (IX) 200238'),  
('10', 'PLAN003', 200304, 'ESTRATEGIA, GESTION Y ADQUISICION DE SI', 'O', '2', '2', '(IX) 200303'),  
('10', 'PLAN003', 200236, 'GERENCIA DE INFORMATICA Y DIRECCION ESTRATEGICA', 'O', '2', '2', '(VIII) 200281 - (IX) 200234'),  
('10', 'PLAN003', 200305, 'GOBIERNO DE TI Y GOBERNANZA DE DATOS', 'O', '2', '2', '(IX) 200284'),  
('10', 'PLAN003', 150303, 'INGLES X', 'O', '0', '2', '(IX) 150300'),  
('10', 'PLAN003', 200241, 'SEMINARIO DE INVESTIGACION 2 INGENIERIA DE SISTEMAS', 'O', '2', '4', '(IX) 200237'),  
('10', 'PLAN003', 200306, 'SISTEMAS DE GESTION DEL CONOCIMIENTO', 'O', '2', '4', '(IX) 200284'),  
('10', 'PLAN003', 200288, 'TECNOLOGIA VERDE', 'O', '2', '0', '(IX) 200285'),  
('10', 'PLAN003', 200287, 'TOPICOS AVANZADOS DE COMPUTACION GRAFICA', 'E', '2', '2', '(VIII) 200153')

;

-- Inserts para la tabla Estudiante

INSERT INTO Estudiante (CodEstudiante, CodCurso, CodProfesion, nombreE, beneficioE, categoriaE)

VALUES

(2022101028, 2000, 'PROF001', 'Villalobos Silva, Khleiwin', '1/4 beca', 'C'),  
(2022101010, 2000, 'PROF001', 'Gómez, Carlos', '1/2 beca', 'A'),  
(2022101011, 2000, 'PROF001', 'Martínez, Ana', '1/4 beca', 'B'),  
(2022101012, 2000, 'PROF001', 'Gomes,Anaceli', 'Regular', 'B'),  
(2022101013, 2000, 'PROF001', 'Crrazco, Martín', '1/2 beca', 'C'),  
(2022101014, 2000, 'PROF001', 'Solis, Grabiel', '1/4 beca', 'C'),  
(2022101015, 2000, 'PROF001', 'Valles, Luis', 'Regular', 'B'),

(2022101028, 2001, 'PROF001', 'Villalobos Silva, Khleiwin', '1/4 beca', 'C'),  
(2022101010, 2001, 'PROF001', 'Gómez, Carlos', '1/2 beca', 'A'),  
(2022101011, 2001, 'PROF001', 'Martínez, Ana', '1/4 beca', 'B'),  
(2022101012, 2001, 'PROF001', 'Gomes,Anaceli', 'Regular', 'B'),

(2022101013, 2001, 'PROF001', 'Crrazco, Martín', '1/2 beca', 'C'),  
(2022101014, 2001, 'PROF001', 'Solis, Grabiel ', '1/4 beca', 'C'),  
(2022101015, 2001, 'PROF001', 'Valles, Luis', 'Regular', 'B'),

(2022101028, 2002, 'PROF001', 'Villalobos Silva, Khleiwin ', '1/4 beca', 'C'),  
(2022101010, 2002, 'PROF001', 'Gómez, Carlos', '1/2 beca', 'A'),  
(2022101011, 2002, 'PROF001', 'Martínez, Ana', '1/4 beca', 'B'),  
(2022101012, 2002, 'PROF001', 'Gomes,Anaceli', 'Regular', 'B'),  
(2022101013, 2002, 'PROF001', 'Crrazco, Martín', '1/2 beca', 'C'),  
(2022101014, 2002, 'PROF001', 'Solis, Grabiel ', '1/4 beca', 'C'),  
(2022101015, 2002, 'PROF001', 'Valles, Luis', 'Regular', 'B'),

(2022101028, 2003, 'PROF001', 'Villalobos Silva, Khleiwin ', '1/4 beca', 'C'),  
(2022101010, 2003, 'PROF001', 'Gómez, Carlos', '1/2 beca', 'A'),  
(2022101011, 2003, 'PROF001', 'Martínez, Ana', '1/4 beca', 'B'),  
(2022101012, 2003, 'PROF001', 'Gomes,Anaceli', 'Regular', 'B'),  
(2022101013, 2003, 'PROF001', 'Crrazco, Martín', '1/2 beca', 'C'),  
(2022101014, 2003, 'PROF001', 'Solis, Grabiel ', '1/4 beca', 'C'),  
(2022101015, 2003, 'PROF001', 'Valles, Luis', 'Regular', 'B'),

(2022101028, 2004, 'PROF001', 'Villalobos Silva, Khleiwin ', '1/4 beca', 'C'),  
(2022101010, 2004, 'PROF001', 'Gómez, Carlos', '1/2 beca', 'A'),  
(2022101011, 2004, 'PROF001', 'Martínez, Ana', '1/4 beca', 'B'),  
(2022101012, 2004, 'PROF001', 'Gomes,Anaceli', 'Regular', 'B'),  
(2022101013, 2004, 'PROF001', 'Crrazco, Martín', '1/2 beca', 'C'),  
(2022101014, 2004, 'PROF001', 'Solis, Grabiel ', '1/4 beca', 'C'),  
(2022101015, 2004, 'PROF001', 'Valles, Luis', 'Regular', 'B'),

(2022101028, 2005, 'PROF001', 'Villalobos Silva, Khleiwin ', '1/4 beca', 'C'),  
(2022101010, 2005, 'PROF001', 'Gómez, Carlos', '1/2 beca', 'A'),  
(2022101011, 2005, 'PROF001', 'Martínez, Ana', '1/4 beca', 'B'),  
(2022101012, 2005, 'PROF001', 'Gomes,Anaceli', 'Regular', 'B'),  
(2022101013, 2005, 'PROF001', 'Crrazco, Martín', '1/2 beca', 'C'),  
(2022101014, 2005, 'PROF001', 'Solis, Grabiel ', '1/4 beca', 'C'),  
(2022101015, 2005, 'PROF001', 'Valles, Luis', 'Regular', 'B')

;

-- Inserts para la tabla Notas  
INSERT INTO Notas (CodEstudiante, CodCurso, p1, p2, p3, ef, ec1, ec2, ec3, ec4, ec5, ev6)  
VALUES  
(2022101028, 2000, 15.5, 16.0, 14.5, 15.8, 16.0, 15.5, 14.0, 16.0, 15.0, 14.5),



;

-- Inserts para la tabla Pago

```
INSERT INTO Pago (CodPago, CodEstudiante, fechaV, numerocuota, monto, descuento)
VALUES
('PAG001', 2022101010, '2024-03-15 00:00:00', '1', 300.00, 'Descuento por pronto pago'),
('PAG002', 2022101010, '2024-04-15 00:00:00', '2', 300.00, 'Descuento por pronto pago'),
('PAG003', 2022101010, '2024-05-15 00:00:00', '3', 337.00, NULL),
('PAG004', 2022101010, '2024-06-15 00:00:00', '4', 300.00, NULL),
('PAG005', 2022101010, '2024-07-10 00:00:00', '5', 300.00, NULL),

('PAG001', 2022101011, '2024-03-15 00:00:00', '1', 450.00, 'Descuento por pronto pago'),
('PAG002', 2022101011, '2024-04-15 00:00:00', '2', 450.00, 'Descuento por pronto pago'),
('PAG003', 2022101011, '2024-05-15 00:00:00', '3', 467.00, NULL),
('PAG004', 2022101011, '2024-06-15 00:00:00', '4', 450.00, NULL),
('PAG005', 2022101011, '2024-07-10 00:00:00', '5', 450.00, NULL),

('PAG001', 2022101012, '2024-03-15 00:00:00', '1', 567.00, 'Descuento por pronto pago'),
('PAG002', 2022101012, '2024-04-15 00:00:00', '2', 567.00, 'Descuento por pronto pago'),
('PAG003', 2022101012, '2024-05-15 00:00:00', '3', 587.00, NULL),
('PAG004', 2022101012, '2024-06-15 00:00:00', '4', 567.00, NULL),
('PAG005', 2022101012, '2024-07-10 00:00:00', '5', 567.00, NULL),

('PAG001', 2022101013, '2024-03-15 00:00:00', '1', 300.00, 'Descuento por pronto pago'),
('PAG002', 2022101013, '2024-04-15 00:00:00', '2', 300.00, 'Descuento por pronto pago'),
('PAG003', 2022101013, '2024-05-15 00:00:00', '3', 337.00, NULL),
('PAG004', 2022101013, '2024-06-15 00:00:00', '4', 300.00, NULL),
('PAG005', 2022101013, '2024-07-10 00:00:00', '5', 300.00, NULL),

('PAG001', 2022101014, '2024-03-15 00:00:00', '1', 450.00, 'Descuento por pronto pago'),
('PAG002', 2022101014, '2024-04-15 00:00:00', '2', 450.00, 'Descuento por pronto pago'),
('PAG003', 2022101014, '2024-05-15 00:00:00', '3', 467.00, NULL),
('PAG004', 2022101014, '2024-06-15 00:00:00', '4', 450.00, NULL),
```

```

('PAG005', 2022101014, '2024-07-10 00:00:00', '5', 450.00, NULL),  

('PAG001', 2022101015, '2024-03-15 00:00:00', '1', 500.00, 'Descuento por pronto pago'),  

('PAG002', 2022101015, '2024-04-15 00:00:00', '2', 500.00, 'Descuento por pronto pago'),  

('PAG003', 2022101015, '2024-05-15 00:00:00', '3', 567.00, NULL),  

('PAG004', 2022101015, '2024-06-15 00:00:00', '4', 585.00, NULL),  

('PAG005', 2022101015, '2024-07-10 00:00:00', '5', 585.00, NULL),  

('PAG001', 2022101028, '2024-03-15 00:00:00', '1', 550.00, 'Descuento por pronto pago'),  

('PAG002', 2022101028, '2024-04-15 00:00:00', '2', 550.00, 'Descuento por pronto pago'),  

('PAG003', 2022101028, '2024-05-15 00:00:00', '3', 567.00, NULL),  

('PAG004', 2022101028, '2024-06-15 00:00:00', '4', 567.00, NULL),  

('PAG005', 2022101028, '2024-07-10 00:00:00', '5', 550.00, NULL)

```

;

-- Inserts para la tabla Planestudios

```

INSERT INTO Planestudios (codPlan, CodProfesion, anioValidacion)  

VALUES  

('PLAN003', 'PROF001', '2022')  

;

```

## SERVICIO NOTAS ServidorNotas

fotos



ADM001.jpg



ADM002.jpg



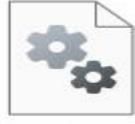
ADM003.jpg



ADM004.jpg



sin\_foto.jpg



Thumbs.db



2022101010.jpg



2022101011.jpg



2022101012.jpg



2022101013.jpg



2022101014.jpg



2022101015.jpg



2022101028.jpg



PROF001.jpg



PROF002.jpg



PROF003.jpg



PROF004.jpg



PROF005.jpg



PROF006.jpg



sin\_foto.jpg



Thumbs.db

## Conexion1.php

```
<?php

class Conexion1 {

    private $cn=null;

    function Conecta() {
        if($this->cn == null) {
            $this->cn = @mysqli_connect("localhost", "root", "", "bdfinalpc3");
        }
        return $this->cn;
    }
}
```

```
}

function getconex() {
    return $this->Conecta();
}


```

```
function Cierra() {
    $this->cn = null;
}

}

?>
```

### Negocio1.php

```
<?php
require_once 'Conexion1.php';
class Negocio1 {
    public function LisNotas($coda){
        $obj=new Conexion1();
        $sql="select c.idcurso, nomcurso, exaparcial, exafinal from Curso c , "
            . "notas n where c.idcurso=n.idcurso and idalumno='$coda' ";
        $res= mysqli_query($obj->Conecta(),$sql) or die(mysqli_error($obj-
>Conecta()));
        $response["dato"]=array();
        while($f= mysqli_fetch_array($res)){
            array_push($response["dato"], array("codc"=>$f[0],"nomc"=>$f[1],
                "ep"=>$f[2],"ef"=>$f[3]));
        }
        return $response;
    }

    public function LisPagos($coda){
        $obj=new Conexion1();
        $sql="select ciclo, fecha, monto from pagos "
            . "where idalumno='$coda' ";
        $res= mysqli_query($obj->Conecta(),$sql) or die(mysqli_error($obj-
>Conecta()));
    }
}
```

```

$response["dato"]=array();
while($f= mysqli_fetch_array($res)){

    array_push($response["dato"], array("cic"=>$f[0],"fec"=>$f[1],
        "mon"=>$f[2]));
}
return $response;
}

public function LisCurso(){
$obj=new Conexion1();
$sql="select idcurso, nomcurso from Curso";
$res= mysqli_query($obj->Conecta(),$sql) or die(mysqli_error($obj-
>Conecta()));
$response["dato"]=array();
while($f= mysqli_fetch_array($res)){
    array_push($response["dato"], array("codc"=>$f[0],"nomc"=>$f[1]));
}
return $response;
}
//LISTA CURSOS DE UIN ALUMNO
public function LisCursoA($codEstudiante){
$obj=new Conexion1();
$sql = "SELECT cp.CodCurso, cp.nombreC
FROM CursoProfesor cp
INNER JOIN Estudiante e ON cp.CodCurso = e.CodCurso
WHERE e.CodEstudiante = '$codEstudiante'";
$res= mysqli_query($obj->Conecta(),$sql) or die(mysqli_error($obj-
>Conecta()));
$response["dato"]=array();
while($f= mysqli_fetch_array($res)){
    array_push($response["dato"], array("codc"=>$f[0],"nomc"=>$f[1]));
}
return $response;
}

//LISTA CURSOS DE UIN profesor
public function LisCursoP($codProfesor){
$obj=new Conexion1();
$sql = "SELECT CodCurso, nombreC from cursoprofesor where
CodProfesor = '$codProfesor'";
$res= mysqli_query($obj->Conecta(),$sql) or die(mysqli_error($obj-
>Conecta()));
$response["dato"]=array();
while($f= mysqli_fetch_array($res)){
    array_push($response["dato"], array("codc"=>$f[0],"nomc"=>$f[1]));
}

```

```

        return $response;
    }
        //LISTA profesores X codAdministrador
    public function LisProfesor($codAdministrador){
        $obj=new Conexion1();
        $sql = "SELECT CodProfesor,CodEspecialidad,nombreP
,telefonoP,direccionP,gradoP,aniosC from profesor where
CodAdministrador='$codAdministrador'";
        $res= mysqli_query($obj->Conecta(),$sql) or die(mysqli_error($obj-
>Conecta()));
        $response["dato"]=array();
        while($f= mysqli_fetch_array($res)){
            array_push($response["dato"],
array("codp"=>$f[0],"code"=>$f[1],"nomp"=>$f[2],"tp"=>$f[3],"dp"=>$f[4],"gp
"=>$f[5],"acp"=>$f[6]));
        }
        return $response;
    }

        //Notas de un curso de un alumno
    public function Boleta($codEstudiante, $codCurso){
        $obj = new Conexion1();
        $sql = "SELECT
            cp.CodCurso AS codc,
            cp.nombreC AS nomc,
            p.nombreP AS nomp,
            n.p1,
            n.p2,
            n.p3,
            n.ef,
            n.ec1 AS c1,
            n.ec2 AS c2,
            n.ec3 AS c3,
            n.ec4 AS c4,
            n.ec5 AS c5,
            n.ev6 AS c6
        FROM
            CursoProfesor cp
        JOIN
            Profesor p ON cp.CodProfesor = p.CodProfesor
        JOIN
            Notas n ON cp.CodCurso = n.CodCurso
        JOIN
            Estudiante e ON n.CodEstudiante = e.CodEstudiante
        WHERE
            e.CodEstudiante = '$codEstudiante' AND
            cp.CodCurso = '$codCurso'";
    }

```

```

$res = mysqli_query($obj->Conecta(), $sql) or die(mysqli_error($obj-
>Conecta()));
$response["dato"] = array();
$codigosCursos = array(); // Array para almacenar los códigos de curso ya
procesados

while ($f = mysqli_fetch_array($res)){
    // Verificar si el código de curso ya ha sido procesado para evitar
    duplicados
    if (!in_array($f['codc'], $codigosCursos)) {
        array_push($response["dato"], array(
            "codc" => $f[0],
            "nomc" => $f[1],
            "nomp" => $f[2],
            "p1" => $f[3],
            "p2" => $f[4],
            "p3" => $f[5],
            "ef" => $f[6],
            "c1" => $f[7],
            "c2" => $f[8],
            "c3" => $f[9],
            "c4" => $f[10],
            "c5" => $f[11],
            "c6" => $f[12]
        ));
        // Agregar el código de curso al array de códigos procesados
        $codigosCursos[] = $f['codc'];
    }
}
return $response;
}


```

```

public function CursosActuales($codEstudiante){
    $obj = new Conexion1();
    $sql = "SELECT
        cp.CodCurso AS codc,
        cp.nombreC AS nomc,
        p.nombreP AS nomp,
        n.p1,
        n.p2,
        n.p3,
        n.ef,
        n.ec1 AS c1,
        n.ec2 AS c2,
        n.ec3 AS c3,

```

```

n.ec4 AS c4,
n.ec5 AS c5,
n.ev6 AS c6
FROM
    CursoProfesor cp
JOIN
    Profesor p ON cp.CodProfesor = p.CodProfesor
JOIN
    Notas n ON cp.CodCurso = n.CodCurso
JOIN
    Estudiante e ON n.CodEstudiante = e.CodEstudiante
WHERE
    e.CodEstudiante = '$codEstudiante';

$res = mysqli_query($obj->Conecta(), $sql) or die(mysqli_error($obj->Conecta()));
$response["dato"] = array();
$codigosCursos = array(); // Array para almacenar los códigos de curso ya procesados

while ($f = mysqli_fetch_array($res)){
    // Verificar si el código de curso ya ha sido procesado para evitar duplicados
    if (!in_array($f['codc'], $codigosCursos)) {
        array_push($response["dato"], array(
            "codc" => $f[0],
            "nomc" => $f[1],
            "nomp" => $f[2],
            "p1" => $f[3],
            "p2" => $f[4],
            "p3" => $f[5],
            "ef" => $f[6],
            "c1" => $f[7],
            "c2" => $f[8],
            "c3" => $f[9],
            "c4" => $f[10],
            "c5" => $f[11],
            "c6" => $f[12]
        ));
        // Agregar el código de curso al array de códigos procesados
        $codigosCursos[] = $f['codc'];
    }
}
return $response;
}

```

//ALUMNOS DE UN CURSO

```
public function NotaAlu($codc){
    $obj=new Conexion1();
    $sql="SELECT a.idalumno, a.apealumno, a.nomalumno, n.exaparcial,
n.exafinal FROM alumno a JOIN notas n ON a.idalumno = n.idalumno
WHERE n.idcurso = '$codc'";
    $res= mysqli_query($obj->Conecta(),$sql) or die(mysqli_error($obj-
>Conecta()));
    $response["dato"]=array();
    while($f= mysqli_fetch_array($res)){

        array_push($response["dato"],
array("coda"=>>$f[0],"ape"=>>$f[1],"nom"=>>$f[2],
"ep"=>>$f[3],"ef"=>>$f[4]));
    }
    return $response;
}
public function pagoAlu($codEstudiante){
    $obj=new Conexion1();
    $sql="select numerocuota, monto,descuento,fechaV from pago where
CodEstudiante=' $codEstudiante' ";
    $res= mysqli_query($obj->Conecta(),$sql) or die(mysqli_error($obj-
>Conecta()));
    $response["dato"]=array();
    while($f= mysqli_fetch_array($res)){

        array_push($response["dato"],
array("cuota"=>>$f[0],"monto"=>>$f[1],"descuento"=>>$f[2],
"fecha"=>>$f[3]));
    }
    return $response;
}

public function login($user,$pass){
    $obj=new Conexion1();
    $sql="select nombre from usuarios "
        . "where user='$user' and pass='$pass' ";
    $res= mysql_query($sql,$obj->Conecta()) or die(mysql_error());

    if (mysql_num_rows($res)>0) {
        $f=mysql_fetch_array($res);
```

```

        $response["dato"]=$f[0];
    }else{
        $response["dato"]="error";
    }
    return $response;
}
//editar notas
public function editarNotas($codEstudiante, $codCurso, $p1, $p2, $p3,
$pef, $c1, $c2, $c3, $c4, $c5, $c6) {
    $obj = new Conexion1();

    // Prevenir SQL Injection usando parámetros preparados
    $sql = "UPDATE notas SET p1=?, p2=?, p3=?, ef=?, ec1=?, ec2=?, ec3=?,
ec4=?, ec5=?, ev6=? WHERE CodEstudiante=? AND CodCurso=?";

    $stmt = mysqli_prepare($obj->getconex(), $sql);
    if ($stmt === false) {
        echo "Error preparando la consulta: " . mysqli_error($obj->getconex());
        return;
    }

    mysqli_stmt_bind_param($stmt, "dddddss", $p1, $p2, $p3, $fef,
$c1, $c2, $c3, $c4, $c5, $c6, $codEstudiante, $codCurso);

    if (!mysqli_stmt_execute($stmt)) {
        echo "Error ejecutando la consulta: " . mysqli_stmt_error($stmt);
    } else {
        echo "Notas actualizadas correctamente.";
    }

    mysqli_stmt_close($stmt);
}

```

```

// Función en PHP para listar estudiantes de un curso
public function LisAluCurso($codCurso) {
    $obj = new Conexion1();
    $sql = "SELECT CodEstudiante, nombreE FROM estudiante WHERE
CodCurso = '$codCurso'";
    $res = mysqli_query($obj->Conecta(), $sql) or die(mysqli_error($obj-
>Conecta()));

    // Inicialización del arreglo de respuesta
    $response["dato"] = array();

    // Iterar sobre los resultados y agregar al arreglo de respuesta

```

```

        while ($f = mysqli_fetch_array($res)) {
            array_push($response["dato"], array("codE" => $f[0], "nomE" => $f[1]));
        }

        return $response;
    }

    public function LisCiclos() {
        $obj = new Conexion1();
        $sql = "SELECT DISTINCT ciclo FROM Detaplan";
        $res = mysqli_query($obj->Conecta(), $sql) or die(mysqli_error($obj->Conecta()));

        // Inicialización del arreglo de respuesta
        $response["dato"] = array();

        // Iterar sobre los resultados y agregar al arreglo de respuesta
        while ($f = mysqli_fetch_array($res)) {
            array_push($response["dato"], array("ciclo" => $f[0]));
        }

        return $response;
    }

    public function DetaPlan($codCiclo){
        $obj=new Conexion1();
        $sql="SELECT codigo, asignatura, tipocurso, ht, hp, prerequisitos
FROM Detaplan WHERE ciclo = $codCiclo";
        $res= mysqli_query($obj->Conecta(),$sql) or die(mysqli_error($obj->Conecta()));
        $response["dato"]=array();
        while($f= mysqli_fetch_array($res)){

            array_push($response["dato"],
array("codc"=>$f[0],"nomc"=>$f[1],"tipo"=>$f[2],
"ht"=>$f[3],"hp"=>$f[4],"req"=>$f[5]));
        }
        return $response;
    }

}

?>
```

## Controla1.php

```
<?php

require_once './Negocio1.php';

$opc=$_REQUEST['tag'];

if ($opc=="consulta1") {

    $coda=$_REQUEST["coda"];

    $obj=new Negocio1();

    echo json_encode($obj->LisNotas($coda));

}

if ($opc=="consulta2") {

    $coda=$_REQUEST["coda"];

    $obj=new Negocio1();

    echo json_encode($obj->LisPagos($coda));

}

if ($opc=="consulta5") {

    $codc=$_REQUEST["codc"];

    $obj=new Negocio1();

    echo json_encode($obj->NotaAlu($codc));

}

if ($opc=="consulta6") {

    $codEstudiante=$_REQUEST["codEstudiante"];

    $obj=new Negocio1();

    echo json_encode($obj->LisCursoA($codEstudiante));

}
```

```
if ($opc=="consulta7") {  
  
    $codProfesor=$_REQUEST["codProfesor"];  
  
    $obj=new Negocio1();  
  
    echo json_encode($obj->LisCursoP($codProfesor));  
  
}  
  
  
  
  
if ($opc=="consulta8") {  
  
    $codAdministrador=$_REQUEST["codAdministrador"];  
  
    $obj=new Negocio1();  
  
    echo json_encode($obj->LisProfesor($codAdministrador));  
  
}  
  
  
  
  
if ($opc == "consulta9") {  
  
    $codEstudiante = $_REQUEST["codEstudiante"];  
  
    $codCurso = $_REQUEST["codCurso"];  
  
    $obj=new Negocio1();  
  
    echo json_encode($obj->Boleta($codEstudiante,$codCurso));  
  
}  
  
  
  
  
if ($opc == "consulta10") {  
  
    $codEstudiante = $_REQUEST["codEstudiante"];  
  
    $obj=new Negocio1();  
  
    echo json_encode($obj->CursosActuales($codEstudiante));
```

```
}

//editar notas

//editar notas

if ($opc == "consulta11") {

    // Asegúrate de utilizar $_POST para obtener los datos

    $codEstudiante = $_POST["codEstudiante"];

    $codCurso = $_POST["codCurso"];

    $p1 = $_POST["p1"];

    $p2 = $_POST["p2"];

    $p3 = $_POST["p3"];

    $ef = $_POST["ef"];

    $c1 = $_POST["c1"];

    $c2 = $_POST["c2"];

    $c3 = $_POST["c3"];

    $c4 = $_POST["c4"];

    $c5 = $_POST["c5"];

    $c6 = $_POST["c6"];


    // Debugging: print received data

    echo "Datos recibidos: codEstudiante=$codEstudiante, codCurso=$codCurso,
p1=$p1, p2=$p2, p3=$p3, ef=$ef, c1=$c1, c2=$c2, c3=$c3, c4=$c4, c5=$c5,
c6=$c6";


    $obj = new Negocio1();

    $obj->editarNotas($codEstudiante, $codCurso, $p1, $p2, $p3, $ef, $c1, $c2,
$c3, $c4, $c5, $c6);
```

```

echo json_encode(["success" => true]);

exit;

}

// alumnos en un curso

if ($opc == "consulta12") {

$codCurso = $_REQUEST["codCurso"];

$obj = new Negocio1();

echo json_encode($obj->LisAluCurso($codCurso));

}

if ($opc == "consulta13") {

$codEstudiante = $_REQUEST["codEstudiante"];

$obj = new Negocio1();

echo json_encode($obj->pagoAlu($codEstudiante));

}

if ($opc=="consulta14") {

$obj=new Negocio1();

echo json_encode($obj->LisCiclos());

}

if ($opc == "consulta15") {

$codCiclo = $_REQUEST["codCiclo"];

```

```
$obj = new Negocio1();

echo json_encode($obj->DetaPlan($codCiclo));

}
```

```
if ($opc=="consulta3") {

$obj=new Negocio1();

echo json_encode($obj->LisCurso());

}

if ($opc=="consulta4") {

$use=$_REQUEST["user"];

$pas=$_REQUEST["pass"];

$obj=new Negocio1();

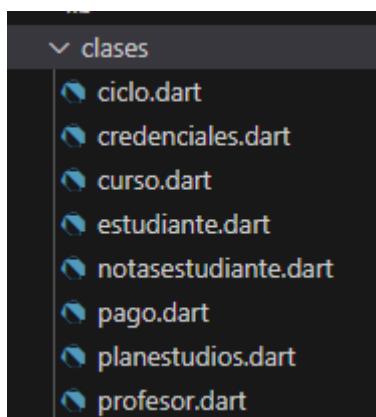
echo json_encode($obj->login($use,$pas));

}

?>
```

## FLUTTER

### Clases



### Ruta.dart

```
class Ruta {  
    static String ip = "ip de maquina";  
}
```

### Ciclo.dart

```
class Ciclo {  
    String cic;  
    Ciclo(  
        {required this.cic}); //al usar las llaves se recibe respuesta en cod json  
}
```

### Credenciales.dart

```
class Credenciales {  
    String user=" ";  
    String password="";  
}
```

### Curso.dart

```
class Curso{  
    String codc;  
    String nomc;  
    Curso({required this.codc,required this.nomc}); //al usar las llaves se recibe  
    respuesta en cod json  
}
```

### Estudiante.dart

```
class Estudiante {  
    int coda = 0;  
    String noma = "";  
}
```

### Notasestudiante.dart

```
class PlanEstudios {  
    final String codc;
```

```
final String nomc;
final String nomp;
double p1;
double p2;
double p3;
double ef;
double c1;
double c2;
double c3;
double c4;
double c5;
double c6;

PlanEstudios({
    required this.codc,
    required this.nomc,
    required this.nomp,
    required this.p1,
    required this.p2,
    required this.p3,
    required this.ef,
    required this.c1,
    required this.c2,
    required this.c3,
    required this.c4,
    required this.c5,
    required this.c6,
});

// Factory method fromJson
factory PlanEstudios.fromJson(Map<String, dynamic> json) {
    return PlanEstudios(
        codc: json['codc'] ?? '',
        nomc: json['nomc'] ?? '',
        nomp: json['nomp'] ?? '',
        p1: double.parse(json['p1'].toString()),
        p2: double.parse(json['p2'].toString()),
        p3: double.parse(json['p3'].toString()),
        ef: double.parse(json['ef'].toString()),
        c1: double.parse(json['c1'].toString()),
        c2: double.parse(json['c2'].toString()),
        c3: double.parse(json['c3'].toString()),
        c4: double.parse(json['c4'].toString()),
        c5: double.parse(json['c5'].toString()),
        c6: double.parse(json['c6'].toString()),
    );
}
```

```

// Método para copiar con nuevos valores
PlanEstudios copyWith({
  double? p1,
  double? p2,
  double? p3,
  double? ef,
  double? c1,
  double? c2,
  double? c3,
  double? c4,
  double? c5,
  double? c6,
}) {
  return PlanEstudios(
    codc: this.codc,
    nomc: this.nomc,
    nomp: this.nomp,
    p1: p1 ?? this.p1,
    p2: p2 ?? this.p2,
    p3: p3 ?? this.p3,
    ef: ef ?? this.ef,
    c1: c1 ?? this.c1,
    c2: c2 ?? this.c2,
    c3: c3 ?? this.c3,
    c4: c4 ?? this.c4,
    c5: c5 ?? this.c5,
    c6: c6 ?? this.c6,
  );
}

double promediocontinuas() {
  return (c1 + c2 + c3 + c4 + c5 + c6) / 6;
}

double promedioCurso() {
  return (p1 * 0.1 +
    p2 * 0.2 +
    p3 * 0.2 +
    promediocontinuas() * 0.2 +
    ef * 0.3);
}

```

### Pago.dart

```

class Pago{
  final String cuota;

```

```

final double monto;
final String descuento;
final String fecha;

Pago(
  required this.cuota,
  required this.monto,
  required this.descuento,
  required this.fecha);

factory Pago.fromJson(Map<String, dynamic> json) {
  return Pago(
    cuota: json['cuota'],
    monto: double.parse(json['monto'].toString()),
    descuento: json['descuento'] ?? '',
    fecha: json['fecha'],
  );
}

double des() {
  switch (descuento) {
    case "Descuento por pronto":
      return 0.10 * monto;
    default:
      return 0;
  }
}

double pagoTotal(List<Pago> pagos) {
  double total = 0;
  for (var pago in pagos) {
    total += pago.monto;
  }
  return total - des();
}

```

### *Planestudios.dart*

```

class PlanEstudios {
  final String codc;
  final String nomc;
  final String tp;
  final String ht;
  final String hp;
  final String req;

```

```

  PlanEstudios({

```

```

        required this.codc,
        required this.nomc,
        required this.tp,
        required this.ht,
        required this.hp,
        required this.req,
    });

// Factory method fromJson
factory PlanEstudios.fromJson(Map<String, dynamic> json) {
    return PlanEstudios(
        codc: json['codc'] ?? '',
        nomc: json['nomc'] ?? '',
        tp: json['tipo'] ?? '',
        ht: json['ht'] ?? '',
        hp: json['hp'] ?? '',
        req: json['req'] ?? '',
    );
}

String cre() {
    return '3';
}
}

```

### *Profesor.dart*

```

class Profesor {
    final String codp;
    final String code;
    final String nomp;
    final String celular;
    final String direccp;
    final String grado;
    final String anios;

    Profesor({
        required this.codp,
        required this.code,
        required this.nomp,
        required this.celular,
        required this.direccp,
        required this.grado,
        required this.anios,
    });

    factory Profesor.fromJson(Map<String, dynamic> json) {

```

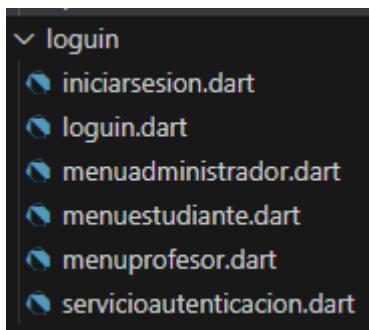
```

return Profesor(
  codp: json['codp'] ?? '',
  code: json['code'] ?? '',
  nomp: json['nomp'] ?? '',
  celular: json['tp'] ?? '',
  direccp: json['dp'] ?? '',
  grado: json['gp'] ?? '',
  anios: json['acp'] ?? '',
);
}

String especialidad(){
  switch (code){
    case "ESP001":
      return "Ingeniería de Software";
    case "ESP002":
      return "Bases de Datos y Programación";
    case "ESP003":
      return "Programación";
    case "ESP004":
      return "Algorítmica";
    case "ESP005":
      return "Ingeniería de Sistemas";
    default:
      return "Estadística";
  }
}
}

```

## Login



### *Iniciarssion.dart*

```

import 'package:flutter/material.dart';
import
'package:flutter_application_notas_remoto/login/menuadministrador.dart';
import 'package:flutter_application_notas_remoto/login/menuestudiante.dart';

```

```

import 'package:flutter_application_notas_remoto/login/login.dart';
import 'package:flutter_application_notas_remoto/login/menuprofesor.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Authentication Example',
      initialRoute: '/login', // Ruta inicial
      routes: {
        '/login': (context) => LoginPage(),
        '/admin_home': (context) => AdministradorHomePage(nombreUsuario: ""),
        '/professor_home': (context) => ProfessorHomePage(nombreUsuario: ""),
        '/estudiante_home': (context) => EstudianteHomePage(nombreUsuario: ""),
      },
    );
  }
}

```

### *Login.dart*

```

import 'package:flutter/material.dart';
import
'package:flutter_application_notas_remoto/login/menuadministrador.dart';
import 'package:flutter_application_notas_remoto/login/menuestudiante.dart';
import 'package:flutter_application_notas_remoto/login/menuprofesor.dart';
import
'package:flutter_application_notas_remoto/login/servicioautenticacion.dart';

class LoginPage extends StatelessWidget {
  final TextEditingController _userController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();

  LoginPage({Key? key}) : super(key: key);

  void _login(BuildContext context) async {
    String user = _userController.text.trim();
    String password = _passwordController.text.trim();

    bool loggedIn = await ServicioCredenciales().login(user, password);

    if (loggedIn) {
      if (password.startsWith('A')) {
        Navigator.pushReplacement(
          context,

```

```
MaterialPageRoute(  
    builder: (context) =>  
        AdministradorHomePage(nombreUsuario: password),  
,  
);  
} else if (password.startsWith('P')) {  
    Navigator.pushReplacement(  
        context,  
        MaterialPageRoute(  
            builder: (context) => ProfessorHomePage(nombreUsuario: password),  
,  
        );  
} else if (user.length == 10) {  
    Navigator.pushReplacement(  
        context,  
        MaterialPageRoute(  
            builder: (context) => EstudianteHomePage(nombreUsuario: user),  
,  
        );  
}  
}  
} else {  
    showDialog(  
        context: context,  
        builder: (BuildContext context) {  
            return AlertDialog(  
                title: Text('Error'),  
                content: Text('Usuario o contraseña incorrectos'),  
                actions: <Widget>[  
                    TextButton(  
                        style: ElevatedButton.styleFrom(  
                            foregroundColor: Color.fromARGB(255, 21, 32, 129),  
                            backgroundColor: Color.fromARGB(255, 61, 168, 187),  
                            padding: EdgeInsets.all(10),  
,  
                        ),  
                        child: Text('Cerrar'),  
                        onPressed: () {  
                            Navigator.of(context).pop();  
                        },  
                    ),  
                ],  
            );  
        },  
    );  
}  
}  
}  
  
@override
```

```
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: Color.fromARGB(255, 25, 37, 206),
      foregroundColor: Color.fromARGB(255, 235, 234, 238),
      title: Text('INICIAR SESIÓN'),
    ),
    body: Padding(
      padding: EdgeInsets.all(16.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          // Widget para mostrar la foto de perfil o una imagen predeterminada
          ImageWidget(userController: _userController),
          SizedBox(height: 20.0),
          TextField(
            controller: _userController,
            decoration: InputDecoration(
              labelText: 'Usuario',
            ),
          ),
          TextField(
            controller: _passwordController,
            decoration: InputDecoration(
              labelText: 'Contraseña',
            ),
            obscureText: true,
          ),
          SizedBox(height: 20.0),
          ElevatedButton(
            style: TextButton.styleFrom(
              foregroundColor: Color.fromARGB(255, 0, 0, 0),
              backgroundColor: Color.fromARGB(255, 61, 126, 187),
              padding: EdgeInsets.all(10),
            ),
            onPressed: () {
              _login(context);
            },
            child: Text('Iniciar sesión'),
          ),
        ],
      ),
    );
}
```

```
class ImageWidget extends StatelessWidget {
final TextEditingController userController;

const ImageWidget({Key? key, required this.userController}) : super(key: key);

@Override
Widget build(BuildContext context) {
String imageUrl =
"http://192.168.42.2/ServidorNotas/fotos1/${userController.text.trim()}.jpg";

return ClipOval(
child: Image.network(
imageUrl,
width: 50,
height: 50,
fit: BoxFit.cover,
errorBuilder:
(BuildContext context, Object exception, StackTrace? stackTrace) {
return Image.asset(
'assets/sin_foto.jpg', // Ruta de la imagen predeterminada
width: 50,
height: 50,
fit: BoxFit.cover,
);
},
),
);
}
}
```

#### Servicioautenticacion.dart

```
class ServicioCredenciales {
Future<bool> login(String user, String password) async {
if ((user == 'administrador' && password == 'ADM001') ||
(user == 'administrador' && password == 'ADM002') ||
(user == 'administrador' && password == 'ADM003') ||
(user == 'administrador' && password == 'ADM004')) {
return true;
} else if ((user == '@ramirez' && password == 'PROF001') ||
(user == '@lopez' && password == 'PROF002') ||
(user == '@quispe' && password == 'PROF003') ||
(user == '@piedra' && password == 'PROF004') ||
(user == '@cardenas' && password == 'PROF005') ||
(user == '@huaman' && password == 'PROF006')) {
return true;
} else if ((user == '2022101028' && password == '1234') ||
```

```

(user == '2022101010' && password == '1234') ||
(user == '2022101011' && password == '1234') ||
(user == '2022101012' && password == '1234') ||
(user == '2022101013' && password == '1234') ||
(user == '2022101014' && password == '1234') ||
(user == '2022101015' && password == '1234')) {
    return true;
} else {
    return false;
}
}
}

```

### *Menuadministrador.dart*

```

import 'package:flutter/material.dart';
import 'package:flutter_application_notas_remoto/pagprofxadministrador.dart';

class AdministradorHomePage extends StatelessWidget {
    final String nombreUsuario;

    AdministradorHomePage({required this.nombreUsuario});

    Future<void> confirmarCerrar(BuildContext context) async {
        return showDialog<void>(
            context: context,
            builder: (BuildContext context) {
                return AlertDialog(
                    title: Text('¿Cerrar sesión?'),
                    content: SingleChildScrollView(
                        child: ListBody(
                            children: <Widget>[
                                Text('¿Está seguro de que desea cerrar la sesión?'),
                            ],
                        ),
                    ),
                    actions: <Widget>[
                        TextButton(
                            child: Text('Cancelar'),
                            onPressed: () {
                                Navigator.of(context).pop();
                            },
                        ),
                        TextButton(
                            child: Text('Cerrar sesión'),
                            onPressed: () {
                                Navigator.of(context).pushReplacementNamed('/login');
                            },
                        ),
                    ],
                );
            }
        );
    }
}

```

```
        },
        ),
        ],
        );
    },
    );
}
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            foregroundColor: Color.fromARGB(255, 21, 32, 129),
            backgroundColor: Color.fromARGB(255, 61, 168, 187),
            title: Text('PANEL ADMINISTRADOR'),
            actions: [
                IconButton(
                    icon: Icon(Icons.exit_to_app),
                    onPressed: () {
                        confirmarCerrar(context);
                    },
                ),
            ],
        ),
        body: Center(
            child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                    Image.network(
                        "http://192.168.42.2/ServidorNotas/fotos/\$nombreUsuario.jpg",
                        errorBuilder: (BuildContext context, Object exception,
                            StackTrace? stackTrace) {
                            return Image.asset('assets/sin_foto.jpg');
                        },
                        width: 200,
                        height: 200,
                        fit: BoxFit.cover,
                    ),
                    Text(
                        'BIENVENIDO ADMINISTRADOR, \$nombreUsuario',
                        style: TextStyle(fontSize: 20),
                    ),
                    SizedBox(height: 20.0),
                    ElevatedButton(
                        style: TextButton.styleFrom(
                            foregroundColor: Color.fromARGB(255, 229, 229, 235),
                            backgroundColor: Color.fromARGB(255, 9, 11, 156),
                        ),
                    ),
                ],
            ),
        ),
    );
}
```

```

        padding: EdgeInsets.all(10),
    ),
    onPressed: () {
        Navigator.push(
            context,
            MaterialPageRoute(
                builder: (context) =>
                    MyAppPxadm(password: nombreUsuario)),
        );
    },
    child: const Text('ADMINISTRAR PROF & ALU'),
),
],
),
),
),
);
}
}

class MyAppPxadm extends StatelessWidget {
final String password;

const MyAppPxadm({Key? key, required this.password}) : super(key: key);

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            foregroundColor: Color.fromARGB(255, 0, 0, 0),
            backgroundColor: Color.fromARGB(255, 118, 119, 197),
            title: Text('INFORMACION DE PROFESORES'),
        ),
        body: Pxadministrador(password: password),
    );
}
}

```

### *Menuprofesor.dart*

```

import 'package:flutter/material.dart';
import 'package:flutter_application_notas_remoto/pagcursosxprof.dart';

class ProfessorHomePage extends StatelessWidget {
final String nombreUsuario;

ProfessorHomePage({required this.nombreUsuario});

```

```
Future<void> confirmarCerrar(BuildContext context) async {
  return showDialog<void>(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: Text('¿Cerrar sesión?'),
        content: SingleChildScrollView(
          child: ListBody(
            children: <Widget>[
              Text('¿Está seguro de que desea cerrar la sesión?'),
            ],
          ),
        ),
        actions: <Widget>[
          TextButton(
            child: Text('Cancelar'),
            onPressed: () {
              Navigator.of(context).pop();
            },
          ),
          TextButton(
            child: Text('Cerrar sesión'),
            onPressed: () {
              Navigator.of(context).pushReplacementNamed('/login');
            },
          ),
        ],
      );
    },
  );
}

@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      foregroundColor: Color.fromARGB(255, 21, 32, 129),
      backgroundColor: Color.fromARGB(255, 61, 168, 187),
      title: Text('PANEL PROFESOR'),
      actions: [
        IconButton(
          icon: Icon(Icons.exit_to_app),
          onPressed: () {
            confirmarCerrar(context);
          },
        ),
      ],
    ),
  );
}
```

```

),
body: Center(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      Image.network(
        "http://192.168.42.2/ServidorNotas/fotos1/$nombreUsuario.jpg",
        errorBuilder: (BuildContext context, Object exception,
          StackTrace? stackTrace) {
          return Image.asset('assets/sin_foto.jpg');
        },
        width: 100,
        height: 100,
        fit: BoxFit.cover,
      ),
      Text(
        'BIENVENIDO PROFESOR, $nombreUsuario',
        style: TextStyle(fontSize: 20),
      ),
      SizedBox(height: 20.0),
      ElevatedButton(
        style: TextButton.styleFrom(
          foregroundColor: Color.fromARGB(255, 229, 229, 235),
          backgroundColor: Color.fromARGB(255, 9, 11, 156),
          padding: EdgeInsets.all(10),
        ),
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) =>
                MyAppcnotasA(password: nombreUsuario)),
          );
        },
        child: const Text('CURSOS Y NOTAS x ALUMNO'),
      ),
    ],
  ),
);
};

}

class MyAppcnotasA extends StatelessWidget {
final String password;

const MyAppcnotasA({Key? key, required this.password}) : super(key: key);

```

```

@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      foregroundColor: Color.fromARGB(255, 241, 241, 240),
      backgroundColor: Color.fromARGB(255, 131, 99, 10),
      title: Text('CURSOS Y NOTAS x ALUMNO'),
    ),
    body: CnotasP(password: password),
  );
}
}

```

### *Menuestudiante.dart*

```

import 'package:flutter/material.dart';
import 'package:flutter_application_notas_remoto/pagboletaCA.dart';
import 'package:flutter_application_notas_remoto/pagplanetstudios.dart';
import 'package:flutter_application_notas_remoto/pagcursosxalu.dart';
import 'package:flutter_application_notas_remoto/pagpagosporalu.dart';

class EstudianteHomePage extends StatelessWidget {
  final String nombreUsuario;

  EstudianteHomePage({required this.nombreUsuario});

  Future<void> confirmarCerrar(BuildContext context) async {
    return showDialog<void>(
      context: context,
      builder: (BuildContext context) {
        return AlertDialog(
          title: Text('¿Cerrar sesión?'),
          content: SingleChildScrollView(
            child: ListBody(
              children: <Widget>[
                Text('¿Está seguro de que desea cerrar la sesión?'),
              ],
            ),
          ),
        ),
        actions: <Widget>[
          TextButton(
            child: Text('Cancelar'),
            onPressed: () {
              Navigator.of(context).pop();
            },
          ),
        ],
      ),
    );
  }
}

```

```
        ),
        TextButton(
            child: Text('Cerrar sesión'),
            onPressed: () {
                Navigator.of(context).pushReplacementNamed('/login');
            },
        ),
    ],
);
},
);
}
}

@Override
Widget build(BuildContext context) {
return Scaffold(
    appBar: AppBar(
        foregroundColor: Color.fromARGB(255, 21, 32, 129),
        backgroundColor: Color.fromARGB(255, 61, 168, 187),
        title: Text('Panel estudiante'),
        actions: [
            IconButton(
                icon: Icon(Icons.exit_to_app),
                onPressed: () {
                    confirmarCerrar(context);
                },
            ),
        ],
),
body: Center(
    child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
            Image.network(
                "http://192.168.42.2/ServidorNotas/fotos1/$nombreUsuario.jpg",
                errorBuilder: (BuildContext context, Object exception,
                    StackTrace? stackTrace) {
                    return Image.asset('assets/sin_foto.jpg');
                },
                width: 100,
                height: 100,
                fit: BoxFit.cover,
            ),
            Text(
                'Bienvenido estudiante, $nombreUsuario',
                style: TextStyle(fontSize: 20),
            ),
        ],
    ),
);
```

```
SizedBox(height: 20.0),
ElevatedButton(
  style: TextButton.styleFrom(
    foregroundColor: Color.fromARGB(255, 229, 229, 235),
    backgroundColor: Color.fromARGB(255, 9, 11, 156),
    padding: EdgeInsets.all(10),
  ),
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) =>
          PagCursosxAlu(password: nombreUsuario)),
    );
  },
  child: const Text('BOLETA POR CURSO'),
),
ElevatedButton(
  style: TextButton.styleFrom(
    foregroundColor: Color.fromARGB(255, 228, 228, 235),
    backgroundColor: Color.fromARGB(255, 131, 12, 47),
    padding: EdgeInsets.all(10),
  ),
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) =>
          pagBoletaCA.PagBoletaCA(password: nombreUsuario)),
    );
  },
  child: const Text('BOLETA NOTAS'),
),
ElevatedButton(
  style: TextButton.styleFrom(
    foregroundColor: Color.fromARGB(255, 229, 229, 235),
    backgroundColor: Color.fromARGB(255, 9, 11, 156),
    padding: EdgeInsets.all(10),
  ),
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) =>
          PagPagosPorAlu(password: nombreUsuario)),
    );
  },
),
```

```
        child: const Text('PAGOS'),
    ),
    ElevatedButton(
        style: TextButton.styleFrom(
            foregroundColor: Color.fromARGB(255, 229, 229, 235),
            backgroundColor: Color.fromARGB(255, 9, 11, 156),
            padding: EdgeInsets.all(10),
        ),
        onPressed: () {
            Navigator.push(
                context,
                MaterialPageRoute(
                    builder: (context) =>
                        PagPlanEstudios(password: nombreUsuario)),
            );
        },
        child: const Text('PLAN ACADÉMICO'),
    ),
],
),
),
),
);
}
}
```

## PAGINAS DE NAVEGACION

● pagadministrarnotasE.dart
● pagadministrarnotasE1.dart
● pagboletaCA.dart
● pagcursosprofesoradm.dart
● pagcursosxalu.dart
● pagcursosxprof.dart
● pagmodificarnotas.dart
● pagpagosporalu.dart
● pagplanestudios.dart
● pagproxadmindador.dart

*pagadministrarnotasE.dart*

```
import 'package:flutter/material.dart';
import 'package:flutter_application_notas_remoto/clases/notasestudiante.dart';
import 'package:flutter_application_notas_remoto/pagcursosxprof.dart';
import 'package:flutter_application_notas_remoto/pagmodificarnotas.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';
```

```
class MyNota extends StatefulWidget {
final String codE;
final String codC;
final String nombreUsuario;

const MyNota(
    {required this.codE,
    required this.codC,
    required this.nombreUsuario,
    Key? key})
    : super(key: key);

@Override
_MyNotaState createState() => _MyNotaState();
}

class _MyNotaState extends State<MyNota> {
List<PlanEstudios> notas = [];

@Override
void initState() {
super.initState();
obtenerNotas();
}

Future<void> obtenerNotas() async {
final url = Uri.parse(
"http://192.168.42.2/ServidorNotas/Controla1.php?tag=consulta9&codEstudiante=${widget.codE}&codCurso=${widget.codC}");

final response = await http.get(url);

if (response.statusCode == 200) {
setState(() {
var resp = jsonDecode(response.body);
List<dynamic> lista = resp["dato"];
notas = lista.map((item) => PlanEstudios.fromJson(item)).toList();
});
} else {
print('Error en la carga de notas');
}
}

@Override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
```

```
backgroundColor: Color.fromRGBO(255, 25, 37, 206),  
foregroundColor: Color.fromRGBO(255, 235, 234, 238),  
title: Text('Notas del Estudiante'),  
,  
body: Center(  
    child: Column(  
        mainAxisAlignment: MainAxisAlignment.center,  
        children: [  
            Text(  
                'Código del Estudiante: ${widget.codE}',  
                style: TextStyle(fontWeight: FontWeight.bold),  
            ),  
            SizedBox(height: 20),  
            Text(  
                'Código del curso: ${widget.codC}',  
                style: TextStyle(fontWeight: FontWeight.bold),  
            ),  
            Expanded(  
                child: ListView.builder(  
                    itemCount: notas.length,  
                    itemBuilder: (BuildContext context, int index) {  
                        return Card(  
                            child: ListTile(  
                                title: Text(  
                                    'Curso: ${notas[index].nomC} (${notas[index].codC})',  
                                    style: TextStyle(fontWeight: FontWeight.bold),  
                                ),  
                                subtitle: Column(  
                                    crossAxisAlignment: CrossAxisAlignment.start,  
                                    children: [  
                                        Text('P1: ${notas[index].p1.toStringAsFixed(2)}'),  
                                        Text('P2: ${notas[index].p2.toStringAsFixed(2)}'),  
                                        Text('P3: ${notas[index].p3.toStringAsFixed(2)}'),  
                                        Text('EF: ${notas[index].ef.toStringAsFixed(2)}'),  
                                        Text('C1: ${notas[index].c1.toStringAsFixed(2)}'),  
                                        Text('C2: ${notas[index].c2.toStringAsFixed(2)}'),  
                                        Text('C3: ${notas[index].c3.toStringAsFixed(2)}'),  
                                        Text('C4: ${notas[index].c4.toStringAsFixed(2)}'),  
                                        Text('C5: ${notas[index].c5.toStringAsFixed(2)}'),  
                                        Text('C6: ${notas[index].c6.toStringAsFixed(2)}'),  
                                        Divider(),  
                                        Text(  
                                            'Promedio Continuas:  
${notas[index].promedioContinuas().toStringAsFixed(2)}'),  
                                        Text(  
                                            'Promedio Curso:  
${notas[index].promedioCurso().toStringAsFixed(2)}'),  
                                    ],  
                                ),  
                            ),  
                        );  
                    },  
                ),  
            ),  
        ],  
    ),  
);
```

```
        ],
      ),
    ),
  );
},
),
),
),
),
ElevatedButton(
  style: TextButton.styleFrom(
    foregroundColor: Color.fromARGB(255, 0, 0, 0),
    backgroundColor: Color.fromARGB(255, 187, 179, 61),
    padding: EdgeInsets.all(10),
  ),
  onPressed: () async {
    final result = await Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => PagModificarNotas(
          codE: widget.codE, codc: widget.codc, notas: notas),
      ),
    );
    if (result != null && result is List<PlanEstudios>) {
      setState(() {
        notas = result;
      });
    }
  },
  child: Text('MODIFICAR NOTAS'),
),
ElevatedButton.icon(
  style: TextButton.styleFrom(
    foregroundColor: Color.fromARGB(255, 0, 0, 0),
    backgroundColor: Color.fromARGB(255, 187, 179, 61),
    padding: EdgeInsets.all(10),
  ),
  onPressed: () async {
    Navigator.pushReplacement(
      context,
      MaterialPageRoute(
        builder: (context) =>
          CnotasP(password: widget.nombreUsuario)),
    );
  },
  icon: Icon(Icons.arrow_back),
  label: Text('RETORNAR'),
),
],
```

```
        ),  
        ),  
    );  
}  
}
```

### *pagadministrarnotasE1.dart*

```
import 'package:flutter/material.dart';  
import 'package:flutter_application_notas_remoto/clases/notasestudiante.dart';  
import 'package:flutter_application_notas_remoto/pagcursosprofesoradm.dart';  
import 'package:flutter_application_notas_remoto/pagmodificarnotas.dart';  
import 'package:http/http.dart' as http;  
import 'dart:convert';  
  
class PagAdministrarNotasE extends StatefulWidget {  
    final String codE;  
    final String codc;  
    final String prof;  
    final String nombreUsuario;  
  
    const PagAdministrarNotasE({  
        required this.codE,  
        required this.codc,  
        required this.nombreUsuario,  
        required this.prof,  
        Key? key,  
    }) : super(key: key);  
  
    @override  
    _PagAdministrarNotasEState createState() => _PagAdministrarNotasEState();  
}  
  
class _PagAdministrarNotasEState extends State<PagAdministrarNotasE> {  
    List<PlanEstudios> notas = [];  
  
    @override  
    void initState(){  
        super.initState();  
        obtenerNotas();  
    }  
  
    Future<void> obtenerNotas() async {  
        final url = Uri.parse(  
            "http://192.168.42.111/ServidorNotas/Controla1.php?tag=consulta9&codEstu  
diante=${widget.codE}&codCurso=${widget.codc}");  
    }
```

```
final response = await http.get(url);

if (response.statusCode == 200) {
  setState(() {
    var resp = jsonDecode(response.body);
    List<dynamic> lista = resp["dato"];
    notas = lista.map((item) => PlanEstudios.fromJson(item)).toList();
  });
} else {
  print('Error en la carga de notas');
}
}

@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: Color.fromRGBO(255, 25, 37, 206),
      foregroundColor: Color.fromRGBO(255, 235, 234, 238),
      title: Text('Notas del Estudiante'),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
            'Código del Estudiante: ${widget.codE}',
            style: TextStyle(fontWeight: FontWeight.bold),
          ),
          SizedBox(height: 20),
          Text(
            'Código del curso: ${widget.codc}',
            style: TextStyle(fontWeight: FontWeight.bold),
          ),
          Expanded(
            child: ListView.builder(
              itemCount: notas.length,
              itemBuilder: (BuildContext context, int index) {
                return Card(
                  child: ListTile(
                    title: Text(
                      'Curso: ${notas[index].nomc} (${notas[index].codc})',
                      style: TextStyle(fontWeight: FontWeight.bold),
                    ),
                    subtitle: Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [

```

```
        Text('P1: ${notas[index].p1.toStringAsFixed(2)}'),
        Text('P2: ${notas[index].p2.toStringAsFixed(2)}'),
        Text('P3: ${notas[index].p3.toStringAsFixed(2)}'),
        Text('EF: ${notas[index].ef.toStringAsFixed(2)}'),
        Text('C1: ${notas[index].c1.toStringAsFixed(2)}'),
        Text('C2: ${notas[index].c2.toStringAsFixed(2)}'),
        Text('C3: ${notas[index].c3.toStringAsFixed(2)}'),
        Text('C4: ${notas[index].c4.toStringAsFixed(2)}'),
        Text('C5: ${notas[index].c5.toStringAsFixed(2)}'),
        Text('C6: ${notas[index].c6.toStringAsFixed(2)})',
        Divider(),
        Text(
            'Promedio Continuas:  

${notas[index].promedioContinuas().toStringAsFixed(2)}'),
        Text(
            'Promedio Curso:  

${notas[index].promedioCurso().toStringAsFixed(2)}'],
        ),
        ),
        );
    },
),
),
ElevatedButton(
    style: TextButton.styleFrom(
        foregroundColor: Color.fromARGB(255, 0, 0, 0),
        backgroundColor: Color.fromARGB(255, 187, 179, 61),
        padding: EdgeInsets.all(10),
    ),
    onPressed: () async {
        final result = await Navigator.push(
            context,
            MaterialPageRoute(
                builder: (context) => PagModificarNotas(
                    codE: widget.codE, codc: widget.codc, notas: notas),
            ),
        );
        if (result != null && result is List<PlanEstudios>) {
            setState(() {
                notas = result;
            });
        }
    },
    child: Text('MODIFICAR NOTAS'),
),
ElevatedButton.icon(
```

```
style: TextButton.styleFrom(
    foregroundColor: Color.fromARGB(255, 0, 0, 0),
    backgroundColor: Color.fromARGB(255, 187, 179, 61),
    padding: EdgeInsets.all(10),
),
 onPressed: () async {
    Navigator.pushReplacement(
        context,
        MaterialPageRoute(
            builder: (context) => PagCursosProfesorAdm(
                codp: widget.prof,
                adminPassword: widget.nombreUsuario,
            )));
},
 icon: Icon(Icons.arrow_back),
 label: Text('RETORNAR'),
),
],
),
),
),
);
);
}
}
```

### *pagboletaCA.dart*

```
import 'package:flutter/material.dart';

import 'package:flutter_application_notas_remoto/clases/notasestudiante.dart';

import
'package:flutter_application_notas_remoto/login/menuadministrador.dart';
import 'package:flutter_application_notas_remoto/login/menuestudiante.dart';
import 'package:flutter_application_notas_remoto/login/menuprofesor.dart';

import 'dart:convert';
import 'package:http/http.dart' as http;

class pagBoletaCA extends StatefulWidget {
final String password;

const pagBoletaCA.PagBoletaCA({Key? key, required this.password})
: super(key: key);

@Override
```

```
_pagBoletaCAState createState() => _pagBoletaCAState();
}

class _pagBoletaCAState extends State<pagBoletaCA> {
List<PlanEstudios> _notas = [];
bool _isLoading = false;

Future<void> _fetchNotas(String password) async {
setState(() {
 isLoading = true;
 _notas = []; // Limpiar resultados anteriores
});

try{
 final response = await http.get(Uri.parse(
 "http://192.168.42.111/ServidorNotas/controla1.php?tag=consulta10&codEst
udiante=$password",
));
if (response.statusCode == 200) {
 final data = json.decode(response.body);
 if (data['dato'] != null && (data['dato'] as List).isNotEmpty) {
 final List<PlanEstudios> fetchedPagos = (data['dato'] as List)
 .map((pago) => PlanEstudios.fromJson(pago))
 .toList();
 setState(() {
 _notas = fetchedPagos;
 });
 } else {
 _showDialog('no hay pagos registrados o pendientes');
 }
 } else {
 print('Error en la carga de datos');
 }
} catch (e) {
 print('Error: $e');
} finally{
 setState(() {
 isLoading = false;
 });
}
}

void _showDialog(String message){
 showDialog(
 context: context,
```

```
builder: (BuildContext context) {
  return AlertDialog(
    title: Text('Información'),
    content: Text(message),
    actions: [
      TextButton(
        child: Text('OK'),
        onPressed: () {
          Navigator.of(context).pop();
        },
      ),
    ],
  );
},
);
}

@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      centerTitle: true,
      foregroundColor: Color.fromARGB(255, 228, 228, 235),
      backgroundColor: Color.fromARGB(255, 131, 12, 47),
      title: Text('BOLETA NOTAS'),
    ),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: [
          SizedBox(height: 16),
          Center(
            child: ElevatedButton(
              style: ElevatedButton.styleFrom(
                foregroundColor: Color.fromARGB(255, 21, 32, 129),
                backgroundColor: Color.fromARGB(255, 61, 168, 187),
                padding: EdgeInsets.all(10),
              ),
              onPressed: () {
                _fetchNotas(widget.password);
              },
              child: Text('VER BOLETA NOTAS'),
            ),
          ),
          SizedBox(height: 16),
          _isLoading
```



```
'P1[10%],  
_notas[index]  
    .p1  
    .toStringAsFixed(2)),  
_buildNoteItem(  
    'P2[20%],  
    _notas[index]  
        .p2  
        .toStringAsFixed(2)),  
_buildNoteItem(  
    'P3[20%],  
    _notas[index]  
        .p3  
        .toStringAsFixed(2)),  
_buildNoteItem(  
    'Ef[30%],  
    _notas[index]  
        .ef  
        .toStringAsFixed(2)),  
],  
),  
Divider(),  
SizedBox(height: 8),  
Row(  
    children: [  
        _buildNoteItem(  
            'C1',  
            _notas[index]  
                .c1  
                .toStringAsFixed(2)),  
        _buildNoteItem(  
            'C2',  
            _notas[index]  
                .c2  
                .toStringAsFixed(2)),  
        _buildNoteItem(  
            'C3',  
            _notas[index]  
                .c3  
                .toStringAsFixed(2)),  
        _buildNoteItem(  
            'C4',  
            _notas[index]  
                .c4  
                .toStringAsFixed(2)),  
    ],  
),
```



```
        nombreUsuario: widget.password,
    ),
),
);
} else if (widget.password.startsWith('P')) {
    Navigator.pushReplacement(
    context,
    MaterialPageRoute(
    builder: (context) => ProfessorHomePage(
        nombreUsuario: widget.password,
    ),
),
);
} else if (widget.password.length == 10) {
    Navigator.pushReplacement(
    context,
    MaterialPageRoute(
    builder: (context) => EstudianteHomePage(
        nombreUsuario: widget.password,
    ),
),
);
}
},
child: Text('MENÚ PRINCIPAL'),
),
],
),
),
],
),
),
);
}
}

Widget _buildNoteItem(String label, String value) {
return Expanded(
    child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: [
            Text(
                '$label',
                style: TextStyle(fontWeight: FontWeight.bold),
            ),
            SizedBox(height: 4),
            Text(value),
        ],
    ),
)
```

```
        ),  
    );  
}  
}
```

### *pagcursosprofesoradm.dart*

```
import 'package:flutter/material.dart';  
import 'package:flutter_application_notas_remoto/clases/curso.dart';  
import 'package:flutter_application_notas_remoto/pagadministrarnotasE1.dart';  
import 'package:http/http.dart' as http;  
import 'dart:convert';  
  
class PagCursosProfesorAdm extends StatefulWidget {  
    final String codp;  
    final String adminPassword;  
  
    const PagCursosProfesorAdm(  
        {Key? key, required this.codp, required this.adminPassword})  
        : super(key: key);  
  
    @override  
    State<PagCursosProfesorAdm> createState() => _PagCursosProfesorAdmState();  
}  
  
class _PagCursosProfesorAdmState extends State<PagCursosProfesorAdm> {  
    List<Curso> doclis = [];  
    List<dynamic> lisnota = [];  
    String? _selectedvalue;  
    String? _selectedCodE;  
  
    // Método para obtener la lista de cursos  
    Future<void> getCurso(String codp) async {  
        final url = Uri.parse(  
            "http://192.168.42.111/ServidorNotas/Controla1.php?tag=consulta7&codProf  
esor=$codp");  
  
        final response = await http.get(url);  
  
        if (response.statusCode == 200) {  
            setState(() {  
                var resp = jsonDecode(response.body);  
                List<dynamic> liscurs = resp["dato"];  
                doclis = liscurs  
                    .map((item) => Curso(codc: item['codc'], nomc: item['nomc']))  
                    .toList();  
            });  
        }  
    }  
}
```

```
    } else {
        print('Error en la carga de datos');
    }
}

// Método para obtener la lista de notas
Future<void> getNotas(String? id) async {
    if (id == null) {
        return;
    }

    final url = Uri.parse(
        "http://192.168.42.111/ServidorNotas/Controla1.php?tag=consulta12&codCu
rso=$id");

    final response = await http.get(url);

    if (response.statusCode == 200) {
        setState(() {
            var resp = jsonDecode(response.body);
            lisnota = resp["dato"];
        });
    } else {
        print('Error en la carga de datos');
    }
}

@Override
void initState() {
    super.initState();
    getCurso(
        widget.codp); // Llama a getCurso con el código del profesor recibido
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            backgroundColor: Color.fromARGB(255, 25, 37, 206),
            foregroundColor: Color.fromARGB(255, 235, 234, 238),
            centerTitle: true,
            title: Text('ALUMNOS POR CURSO'),
        ),
        body: Center(
            child: Column(
                children: [
                    DropdownButton<String>(
```

```

value: _selectedvalue,
hint: Text('Seleccione curso'),
onChanged: (valor) {
    setState(() {
        _selectedvalue = valor!;
        getNotas(_selectedvalue);
    });
},
items: doclis.map((item) {
    return DropdownMenuItem<String>(
        value: item.codc,
        child: Text(item.nomc),
    );
}).toList(),
),
if (_selectedvalue != null && _selectedvalue!.isNotEmpty)
DropdownButton<String>(
    value: _selectedCodE,
    hint: Text('Seleccione COD estudiante'),
    onChanged: (valor) {
        setState(() {
            _selectedCodE = valor!;
        });
    },
    items: lisnota.map((item) {
        return DropdownMenuItem<String>(
            value: item['codE'],
            child: Text(item['codE']),
        );
    }).toList(),
),
SizedBox(height: 16.0),
if (_selectedCodE != null && _selectedCodE!.isNotEmpty)
ElevatedButton.icon(
    icon: Icon(Icons.notes),
    label: Text('Gestionar notas'),
    onPressed: () {
        Navigator.push(
            context,
            MaterialPageRoute(
                builder: (context) => PagAdministrarNotasE(
                    codE: _selectedCodE!,
                    codc: _selectedvalue!,
                    nombreUsuario: widget.adminPassword,
                    prof: widget.codp,
                ),
            ),
        );
    },
),

```

```

        );
    },
),
],
),
),
floatingActionButton: FloatingActionButton(
 onPressed: () {
 Navigator.pop(context);
 },
 child: Icon(Icons.arrow_back),
 backgroundColor: Color.fromARGB(255, 25, 37, 206),
),
);
}
}
}

```

### *pagcursosxalu.dart*

```

import 'package:flutter/material.dart';
import 'package:flutter_application_notas_remoto/clases/curso.dart';
import 'package:flutter_application_notas_remoto/clases/notasestudiante.dart';
import
'package:flutter_application_notas_remoto/login/menuadministrador.dart';
import 'package:flutter_application_notas_remoto/login/menuestudiante.dart';
import 'package:flutter_application_notas_remoto/login/menuprofesor.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';

class PagCursosxAlu extends StatefulWidget {
 final String password;

 const PagCursosxAlu({Key? key, required this.password}) : super(key: key);

 @override
 State<PagCursosxAlu> createState() => _PagCursosxAluState();
}

class _PagCursosxAluState extends State<PagCursosxAlu> {
 List<Curso> doclis = [];
 List<PlanEstudios> lisnota = [];
 String? _selectedCurso;

 @override
 void initState() {
 super.initState();
 getCurso(widget.password); // Cargar lista de cursos al iniciar la página
}

```

```

}

// Obtener lista de cursos para el estudiante
Future<void> getCurso(String password) async {
final response = await http.get(
  Uri.parse(
    "http://192.168.42.2/ServidorNotas/Controla1.php?tag=consulta6&codEstud
iante=$password"),
);

if (response.statusCode == 200) {
  setState(() {
    var resp = jsonDecode(response.body);
    List<dynamic> liscurso = resp["dato"];
    doclis = liscurso
      .map((item) => Curso(codc: item['codc'], nomc: item['nomc']))
      .toList();
    // Actualizar notas al seleccionar el primer curso automáticamente
    if (doclis.isNotEmpty) {
      _selectedCurso = doclis[0].codc;
      // Llamar a getNotas aquí para mostrar las notas del primer curso
      // automáticamente
      getNotas(widget.password, _selectedCurso!);
    }
  });
} else {
  print('Error en la carga de cursos: ${response.statusCode}');
}
}

// Obtener notas del estudiante para el curso seleccionado
Future<void> getNotas(String codEstudiante, String codCurso) async {
final response = await http.get(
  Uri.parse(
    "http://192.168.42.2/ServidorNotas/Controla1.php?tag=consulta9&codEstudi
ante=$codEstudiante&codCurso=$codCurso",
  ),
);

if (response.statusCode == 200) {
  try{
    var jsonResponse = jsonDecode(response.body);
    if (jsonResponse['dato'] != null && jsonResponse['dato'].length > 0) {
      List<dynamic> resp = jsonResponse['dato'];
      List<PlanEstudios> notasFiltradas =
        resp.map((json) => PlanEstudios.fromJson(json)).toList();
    }
  }
}
}

```

```
        setState(() {
            lisnota = notasFiltradas;
        });
    } else {
        print('La respuesta no contiene datos válidos de notas');
    }
} catch (e) {
    print('Error al decodificar JSON: $e');
}
} else {
    print('Error en la carga de notas: ${response.statusCode}');
}
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            backgroundColor: Color.fromRGBO(255, 187, 219, 165),
            foregroundColor: Color.fromRGBO(255, 61, 12, 209),
            centerTitle: true,
            title: Text('Cursos Estudiante'),
        ),
        body: Padding(
            padding: const EdgeInsets.all(16.0),
            child: Column(
                mainAxisAlignment: MainAxisAlignment.start,
                children: [
                    DropdownButton<String>(
                        value: _selectedCurso,
                        hint: Text(
                            'Seleccione curso',
                            style: TextStyle(color: Colors.black),
                        ),
                        onChanged: (newValue) {
                            setState(() {
                                _selectedCurso = newValue!;
                                getNotas(widget.password, _selectedCurso!);
                            });
                        },
                    ),
                    items: doclis.map((Curso curso) {
                        return DropdownMenuItem<String>(
                            value: curso.codc,
                            child: Text(
                                '${curso.codc} - ${curso.nomc}',
                                style: TextStyle(color: Colors.black),
                            ),
                        );
                    })
                ],
            ),
        ),
    );
}
```

```
    );
  }).toList(),
),
SizedBox(height: 16),
Expanded(
  child: ListView.builder(
    itemCount: lisnota.length,
    itemBuilder: (BuildContext context, int index) {
      return Container(
        margin: EdgeInsets.only(bottom: 16.0),
        decoration: BoxDecoration(
          border: Border.all(
            color: Color.fromARGB(255, 51, 3, 224), width: 1.0),
          borderRadius: BorderRadius.circular(8.0),
        ),
        child: Card(
          child: Padding(
            padding: const EdgeInsets.all(16.0),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                ListTile(
                  title: Text('user'),
                  leading: ClipOval(
                    child: Image.network(
                      "http://192.168.42.2/ServidorNotas/fotos1/${widget.password}.jpg",
                    ),
                    errorBuilder: (BuildContext context,
                      Object exception,
                      StackTrace? stackTrace) {
                      return Image.asset('assets/sin_foto.jpg');
                    },
                    width: 50,
                    height: 50,
                    fit: BoxFit.cover,
                  ),
                ),
              ],
            ),
            Divider(),
            SizedBox(height: 8),
            Text(
              'Curso:${doclis[index].nomc} - ${doclis[index].codc}',
              style: TextStyle(fontWeight: FontWeight.bold),
            ),
            Divider(),
            SizedBox(height: 16),
            Text(

```

```
'Prof: ${lisnota[index].nomp}',  
style: TextStyle(fontWeight: FontWeight.bold),  
,  
SizedBox(height: 8),  
Divider(),  
Row(  
children: [  
_buildNoteItem(  
'EC[20%]',  
lisnota[index]  
.promediocontinuas()  
.toStringAsFixed(2)),  
_buildNoteItem('P1[10%]',  
lisnota[index].p1.toStringAsFixed(2)),  
_buildNoteItem('P2[20%]',  
lisnota[index].p2.toStringAsFixed(2)),  
_buildNoteItem('P3[20%]',  
lisnota[index].p3.toStringAsFixed(2)),  
_buildNoteItem('Ef[30%]',  
lisnota[index].p3.toStringAsFixed(2)),  
],  
),  
Divider(),  
SizedBox(height: 8),  
Row(  
children: [  
_buildNoteItem(  
'C1', lisnota[index].c1.toStringAsFixed(2)),  
_buildNoteItem(  
'C2', lisnota[index].c2.toStringAsFixed(2)),  
_buildNoteItem(  
'C3', lisnota[index].c3.toStringAsFixed(2)),  
_buildNoteItem(  
'C4', lisnota[index].c4.toStringAsFixed(2)),  
],  
),  
Row(  
children: [  
_buildNoteItem(  
'C5', lisnota[index].c5.toStringAsFixed(2)),  
_buildNoteItem(  
'C6', lisnota[index].c6.toStringAsFixed(2)),  
],  
),  
SizedBox(height: 16),  
Divider(),  
SizedBox(height: 8),
```

```
        Text(
            'Promedio: ${lisnota[index].promedioCurso().toStringAsFixed(2)}',
            ],
            ),
            ),
            ),
            );
        },
    ),
    ),
    ),
    );
},
),
),
Center(
child: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [
SizedBox(height: 20),
ElevatedButton(
style: TextButton.styleFrom(
foregroundColor: Color.fromARGB(255, 0, 0, 0),
backgroundColor: Color.fromARGB(255, 82, 187, 61),
padding: EdgeInsets.all(10),
),
 onPressed: () {
// Volver a la pantalla de inicio según el tipo de usuario
if (widget.password.startsWith('A')) {
Navigator.pushReplacement(
context,
MaterialPageRoute(
builder: (context) => AdministradorHomePage(
nombreUsuario: widget.password,
),
),
);
} else if (widget.password.startsWith('P')) {
Navigator.pushReplacement(
context,
MaterialPageRoute(
builder: (context) => ProfessorHomePage(
nombreUsuario: widget.password,
),
),
);
} else if (widget.password.length == 10) {
Navigator.pushReplacement(
context,
MaterialPageRoute(
builder: (context) => EstudianteHomePage(
nombreUsuario: widget.password,
),
```

```

),
),
);
}
},
child: Text('MENÚ PRINCIPAL'),
),
],
),
),
],
),
),
),
);
}
}
}

Widget _buildNoteItem(String label, String value) {
return Expanded(
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Text(
'$label',
style: TextStyle(fontWeight: FontWeight.bold),
),
SizedBox(height: 4),
Text(value),
],
),
);
}
}
}

```

### *pagcursosxprof.dart*

```

import 'package:flutter/material.dart';
import 'package:flutter_application_notas_remoto/clases/curso.dart';
import
'package:flutter_application_notas_remoto/loguin/menuadministrador.dart';
import 'package:flutter_application_notas_remoto/loguin/menuestudiante.dart';
import 'package:flutter_application_notas_remoto/loguin/menuprofesor.dart';
import 'package:flutter_application_notas_remoto/pagadmindistrarnotasE.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';

class CnotasP extends StatefulWidget {
final String password;

```

```
const CnotasP({Key? key, required this.password}) : super(key: key);

@Override
State<CnotasP> createState() => _CnotasPState();
}

class _CnotasPState extends State<CnotasP>{
List<Curso> doclis = [];
List<dynamic> lisnota = [];
String _selectedvalue = "";
String _selectedCodE = "";

// Método para obtener la lista de cursos
Future<void> getCurso(String password) async {
final url = Uri.parse(
"http://192.168.42.2/ServidorNotas/Controla1.php?tag=consulta7&codProfesor=$password");

final response = await http.get(url);

if (response.statusCode == 200) {
setState(() {
var resp = jsonDecode(response.body);
List<dynamic> liscurso = resp["dato"];
doclis = liscurso
.map((item) => Curso(codc: item['codc'], nomc: item['nomc']))
.toList();
});
} else {
print('Error en la carga de datos');
}
}

// Método para obtener la lista de notas
Future<void> getNotas(String id) async {
final url = Uri.parse(
"http://192.168.42.2/ServidorNotas/Controla1.php?tag=consulta12&codCurso=$id");

final response = await http.get(url);

if (response.statusCode == 200) {
setState(() {
var resp = jsonDecode(response.body);
lisnota = resp["dato"];
});
}
```





```
Navigator.pushReplacement(
  context,
  MaterialPageRoute(
    builder: (context) => AdministradorHomePage(
      nombreUsuario: widget.password,
    ),
  ),
),
);
} else if (widget.password.startsWith('P')) {
  Navigator.pushReplacement(
    context,
    MaterialPageRoute(
      builder: (context) => ProfessorHomePage(
        nombreUsuario: widget.password,
      ),
    ),
  );
}
} else if (widget.password.length == 10) {
  Navigator.pushReplacement(
    context,
    MaterialPageRoute(
      builder: (context) => EstudianteHomePage(
        nombreUsuario: widget.password,
      ),
    ),
  );
}
},
child: Text('MENÚ PRINCIPAL'),
),
],
),
),
);
}
}
}
```

*pagmodificarnotas.dart*

```
import 'package:flutter/material.dart';
import 'package:flutter_application_notas_remoto/clases/notasestudiante.dart';
import 'package:http/http.dart' as http;

class PagModificarNotas extends StatefulWidget {
  final String codE;
  final String codc;
  final List<PlanEstudios> notas;
```

```
const PagModificarNotas({  
    required this.codE,  
    required this.codC,  
    required this.notas,  
    Key? key,  
}) : super(key: key);  
  
@override  
_PagModificarNotasState createState() => _PagModificarNotasState();  
}  
  
class _PagModificarNotasState extends State<PagModificarNotas> {  
    late List<PlanEstudios> notas;  
    final Map<int, Map<String, TextEditingController>> controllers = {};  
  
    @override  
    void initState() {  
        super.initState();  
        // Crear una copia de las notas recibidas  
        notas = widget.notas.map((nota) => nota.copyWith()).toList();  
  
        // Inicializar los controladores para cada campo  
        for (int i = 0; i < notas.length; i++) {  
            controllers[i] = {  
                'p1': TextEditingController(text: notas[i].p1.toString()),  
                'p2': TextEditingController(text: notas[i].p2.toString()),  
                'p3': TextEditingController(text: notas[i].p3.toString()),  
                'ef': TextEditingController(text: notas[i].ef.toString()),  
                'c1': TextEditingController(text: notas[i].c1.toString()),  
                'c2': TextEditingController(text: notas[i].c2.toString()),  
                'c3': TextEditingController(text: notas[i].c3.toString()),  
                'c4': TextEditingController(text: notas[i].c4.toString()),  
                'c5': TextEditingController(text: notas[i].c5.toString()),  
                'c6': TextEditingController(text: notas[i].c6.toString()),  
            };  
        }  
    }  
  
    Future<void> guardarNotas() async {  
        final url = Uri.parse(  
            "http://192.168.42.2/ServidorNotas/Controla1.php?tag=consulta11&codEstud  
        iante=${widget.codE}&codCurso=${widget.codC}");  
  
        for (var nota in notas) {  
            final response = await http.post(url, body: {  
                'codEstudiante': widget.codE,  
            });  
        }  
    }  
}
```

```
'codCurso': widget.codc,
'p1': nota.p1.toString(),
'p2': nota.p2.toString(),
'p3': nota.p3.toString(),
'ef': nota.ef.toString(),
'c1': nota.c1.toString(),
'c2': nota.c2.toString(),
'c3': nota.c3.toString(),
'c4': nota.c4.toString(),
'c5': nota.c5.toString(),
'c6': nota.c6.toString(),
});

if (response.statusCode != 200) {
  print('Error al guardar las notas: ${response.body}');
} else{
  print('Respuesta del servidor: ${response.body}');
}

// Navegar de regreso a la pantalla anterior y pasar los datos actualizados
Navigator.pop(context, notas);
}

Widget buildNotaTextField(int index, String campo) {
return TextField(
  keyboardType: TextInputType.number,
  decoration: InputDecoration(
    labelText: campo.toUpperCase(),
  ),
  controller: controllers[index]?[campo],
  onChanged: (text) {
    // Actualizar el valor en la lista de notas
    setState(() {
      double? newValue = double.tryParse(text);
      if (newValue != null){
        switch (campo){
          case 'p1':
            notas[index] = notas[index].copyWith(p1: newValue);
            break;
          case 'p2':
            notas[index] = notas[index].copyWith(p2: newValue);
            break;
          case 'p3':
            notas[index] = notas[index].copyWith(p3: newValue);
            break;
          case 'ef':

```

```
        notas[index] = notas[index].copyWith(ef: newValue);
        break;
    case 'c1':
        notas[index] = notas[index].copyWith(c1: newValue);
        break;
    case 'c2':
        notas[index] = notas[index].copyWith(c2: newValue);
        break;
    case 'c3':
        notas[index] = notas[index].copyWith(c3: newValue);
        break;
    case 'c4':
        notas[index] = notas[index].copyWith(c4: newValue);
        break;
    case 'c5':
        notas[index] = notas[index].copyWith(c5: newValue);
        break;
    case 'c6':
        notas[index] = notas[index].copyWith(c6: newValue);
        break;
    default:
        break;
    }
}
});
},
);
}

@Override
void dispose() {
// Liberar los controladores
controllers.forEach((_, map) {
map.forEach((_, controller) {
controller.dispose();
});
});
super.dispose();
}

@Override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
backgroundColor: const Color.fromARGB(255, 25, 37, 206),
foregroundColor: const Color.fromARGB(255, 235, 234, 238),
title: const Text('Modificar Notas del Estudiante'),
```

```
),
body: Center(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Text(
        'Código del Estudiante: ${widget.codE}',
        style: const TextStyle(fontWeight: FontWeight.bold),
      ),
      const SizedBox(height: 20),
      Text(
        'Código del curso: ${widget.codC}',
        style: const TextStyle(fontWeight: FontWeight.bold),
      ),
      Expanded(
        child: ListView.builder(
          itemCount: notas.length,
          itemBuilder: (BuildContext context, int index) {
            return Card(
              child: ListTile(
                title: Text(
                  'Curso: ${notas[index].nomC} (${notas[index].codC})',
                  style: const TextStyle(fontWeight: FontWeight.bold),
                ),
                subtitle: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    buildNotaTextField(index, 'p1'),
                    buildNotaTextField(index, 'p2'),
                    buildNotaTextField(index, 'p3'),
                    buildNotaTextField(index, 'ef'),
                    buildNotaTextField(index, 'c1'),
                    buildNotaTextField(index, 'c2'),
                    buildNotaTextField(index, 'c3'),
                    buildNotaTextField(index, 'c4'),
                    buildNotaTextField(index, 'c5'),
                    buildNotaTextField(index, 'c6'),
                    const SizedBox(height: 10),
                    Text(
                      'Promedio Continuas: ${notas[index].promedioContinuas()}'),
                    Text(
                      'Promedio del Curso: ${notas[index].promedioCurso()}'),
                  ],
                ),
              ),
            );
          },
        ),
      );
    ],
  ),
);
```

```

),
),
ElevatedButton(
  style: TextButton.styleFrom(
    foregroundColor: Color.fromARGB(255, 229, 229, 235),
    backgroundColor: Color.fromARGB(255, 9, 11, 156),
    padding: EdgeInsets.all(10),
  ),
  onPressed: guardarNotas,
  child: const Text('Guardar Notas'),
),
],
),
),
),
);
}
}

```

### *pagpagosporalu.dart*

```

import 'package:flutter/material.dart';
import 'package:flutter_application_notas_remoto/clases/pago.dart';
import
'package:flutter_application_notas_remoto/login/menuadministrador.dart';
import 'package:flutter_application_notas_remoto/login/menuestudiante.dart';
import 'package:flutter_application_notas_remoto/login/menuprofesor.dart';
import 'dart:convert';
import 'package:http/http.dart' as http;

class PagPagosPorAlu extends StatefulWidget {
  final String password; // Recibe el password

  PagPagosPorAlu({required this.password});

  @override
  _PagPagosPorAluState createState() => _PagPagosPorAluState();
}

class _PagPagosPorAluState extends State<PagPagosPorAlu> {
  List<Pago> _pagos = [];
  bool _isLoading = false;

  @override
  void initState() {
    super.initState();
    _fetchPagos(widget.password); // Carga los pagos al inicializar
  }
}

```

```
Future<void> _fetchPagos(String password) async {
  setState(() {
    _isLoading = true;
    _pagos = []; // Limpiar resultados anteriores
  });

  try {
    final response = await http.get(Uri.parse(
      "http://192.168.42.2/ServidorNotas/Controla1.php?tag=consulta13&codEstud
iante=$password",
    ));
    if (response.statusCode == 200) {
      final data = json.decode(response.body);
      if (data['dato'] != null && (data['dato'] as List).isNotEmpty) {
        final List<Pago> fetchedPagos = (data['dato'] as List)
          .map(( pago) => Pago.fromJson(pago))
          .toList();
        setState(() {
          _pagos = fetchedPagos;
        });
      } else {
        _showDialog('El alumno no tiene pagos registrados');
      }
    } else {
      print('Error en la carga de datos');
    }
  } catch (e) {
    print('Error: $e');
  } finally{
    setState(() {
      _isLoading = false;
    });
  }
}

void _showDialog(String message){
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: Text('Información'),
        content: Text(message),
        actions: [
          TextButton(

```



```
        ),
        SizedBox(height: 8),
        Text(
            'Monto: ${pago.monto.toStringAsFixed(2)}'),
        SizedBox(height: 8),
        Text(
            'Descuento considerado: ${pago.des().toStringAsFixed(2)}'),
        Text(
            'Fecha vencimiento: ${pago.fecha}'),
        ],
    ),
),
),
),
),
SizedBox(height: 8),
Text(
    'Pago total: ${_pagos.isEmpty ? 0 :
_pagos[0]. pagoTotal(_pagos).toStringAsFixed(2)}',
    ],
),
),
),
),
),
Center(
    child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
            SizedBox(height: 20),
            ElevatedButton(
                style: TextButton.styleFrom(
                    foregroundColor: Color.fromARGB(255, 0, 0, 0),
                    backgroundColor: Color.fromARGB(255, 82, 187, 61),
                    padding: EdgeInsets.all(10),
                ),
                onPressed: () {
// Volver a la pantalla de inicio según el tipo de usuario
if (widget.password.startsWith('A')) {
                    Navigator.pushReplacement(
                        context,
                        MaterialPageRoute(
                            builder: (context) => AdministradorHomePage(
                                nombreUsuario: widget.password,
                            ),
                        ),
                    );
                } else if (widget.password.startsWith('P')) {
                    Navigator.pushReplacement(
                        context,
                        MaterialPageRoute(

```

```

        builder: (context) => ProfessorHomePage(
            nombreUsuario: widget.password,
        ),
    ),
);
} else if (widget.password.length == 10) {
    Navigator.pushReplacement(
        context,
        MaterialPageRoute(
            builder: (context) => EstudianteHomePage(
                nombreUsuario: widget.password,
            ),
        ),
    );
}
],
),
),
),
],
),
),
),
);
}
}
}

```

### *pagplanestudios.dart*

```

import 'package:flutter/material.dart';
import 'package:flutter_application_notas_remoto/clases/ciclo.dart';
import 'package:flutter_application_notas_remoto/clases/planestudios.dart';
import
'package:flutter_application_notas_remoto/login/menuadministrador.dart';
import 'package:flutter_application_notas_remoto/login/menuestudiante.dart';
import 'package:flutter_application_notas_remoto/login/menuprofesor.dart';

import 'package:http/http.dart' as http;
import 'dart:convert';

class PagPlanEstudios extends StatefulWidget {
    final String password;

    const PagPlanEstudios({Key? key, required this.password}) : super(key: key);

    @override

```

```
State<PagPlanEstudios> createState() => _PagPlanEstudiosState();
}

class _PagPlanEstudiosState extends State<PagPlanEstudios>{
List<Ciclo> doclis = [];
List<PlanEstudios> lisnota = [];
String? _selectedCiclo;

@Override
void initState(){
super.initState();
getCiclo(); // Cargar lista de ciclos al iniciar la página
}

// Obtener lista de ciclos
Future<void> getCiclo() async {
final response = await http.get(
Uri.parse(
"http://192.168.42.2/ServidorNotas/Controla1.php?tag=consulta14"),
);

if (response.statusCode == 200) {
setState(() {
var resp = jsonDecode(response.body);
List<dynamic> lisciclo = resp["dato"];
doclis = lisciclo.map((item) => Ciclo(cic: item['ciclo'])).toList();
// Actualizar notas al seleccionar el primer ciclo automáticamente
if (doclis.isNotEmpty) {
_selectedCiclo = doclis[0].cic;
getPlan(_selectedCiclo!);
}
});
} else {
print('Error en la carga de ciclos: ${response.statusCode}');
}
}

// Obtener datos del plan de estudios para el ciclo seleccionado
Future<void> getPlan(String codCiclo) async {
final response = await http.get(
Uri.parse(
"http://192.168.42.2/ServidorNotas/Controla1.php?tag=consulta15&codCiclo=$codCiclo"),
);

if (response.statusCode == 200) {
try {
```

```
var jsonResponse = jsonDecode(response.body);
if (jsonResponse['dato'] != null && jsonResponse['dato'].length > 0) {
  List<dynamic> resp = jsonResponse['dato'];
  List<PlanEstudios> planes =
    resp.map((json) => PlanEstudios.fromJson(json)).toList();

  setState(() {
    lisnota = planes;
  });
} else {
  print('La respuesta no contiene datos válidos de planes de estudio');
}
} catch (e) {
  print('Error al decodificar JSON: $e');
}
} else {
  print('Error en la carga de datos: ${response.statusCode}');
}
}

@Override
Widget build(BuildContext context) {
return Scaffold(
  appBar: AppBar(
    foregroundColor: Color.fromARGB(255, 0, 0, 0),
    backgroundColor: Color.fromARGB(255, 174, 187, 61),
    title: Text('Plan de Estudios'),
  ),
  body: SingleChildScrollView(
    child: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: [
          DropdownButton<String>(
            value: _selectedCiclo,
            hint: Text('Seleccione ciclo'),
            onChanged: (newValue) {
              setState(() {
                _selectedCiclo = newValue!;
                getPlan(_selectedCiclo!);
              });
            },
            items: doclis.map((Ciclo ciclo) {
              return DropdownMenuItem<String>(
                value: ciclo.cic,
                child: Text('${ciclo.cic}'),
              );
            })
          )
        ],
      ),
    ),
  ),
}
```

```

    );
    }).toList(),
),
SizedBox(height: 16),
lisnota.isEmpty
? Center(
    child: Text('Seleccione un ciclo para cargar los cursos'))
: ListView.builder(
    shrinkWrap: true,
    physics: NeverScrollableScrollPhysics(),
    itemCount: lisnota.length,
    itemBuilder: (BuildContext context, int index) {
        return Card(
            margin: EdgeInsets.symmetric(vertical: 8.0),
            child: Padding(
                padding: const EdgeInsets.all(16.0),
                child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                        Text(
                            'Código: ${lisnota[index].codc}',
                            style: TextStyle(fontWeight: FontWeight.bold),
                        ),
                        SizedBox(height: 8),
                        Text(
                            'Asignatura: ${lisnota[index].nomc}',
                            style: TextStyle(fontWeight: FontWeight.bold),
                        ),
                        SizedBox(height: 8),
                        Text('Tipo: ${lisnota[index].tp}'),
                        SizedBox(height: 8),
                        Text('HT: ${lisnota[index].ht}'),
                        SizedBox(height: 8),
                        Text('HP: ${lisnota[index].hp}'),
                        SizedBox(height: 8),
                        Text('CRD: ${lisnota[index].cre()}'),
                        SizedBox(height: 8),
                        Text('Prerrequisitos: ${lisnota[index].req}'),
                    ],
                ),
            ),
        );
    },
),
SizedBox(height: 16),
Text(
    'INDICACIONES\n'
)

```

```
'- TIPO: Asignatura Obligatoria (O) o Electiva (E).\n'
'- HT: HORAS DE TEORIA requeridas por la asignatura.\n'
'- HP: HORAS DE PRACTICA requeridas por la asignatura.\n'
'- CRD: CRÉDITOS correspondientes a la asignatura.',
style: TextStyle(fontWeight: FontWeight.bold),
),
Center(
child: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [
SizedBox(height: 20),
ElevatedButton(
style: TextButton.styleFrom(
foregroundColor: Color.fromARGB(255, 0, 0, 0),
backgroundColor: Color.fromARGB(255, 82, 187, 61),
padding: EdgeInsets.all(10),
),
 onPressed: () {
// Volver a la pantalla de inicio según el tipo de usuario
if (widget.password.startsWith('A')) {
Navigator.pushReplacement(
context,
MaterialPageRoute(
builder: (context) => AdministradorHomePage(
nombreUsuario: widget.password,
),
),
),
);
} else if (widget.password.startsWith('P')) {
Navigator.pushReplacement(
context,
MaterialPageRoute(
builder: (context) => ProfessorHomePage(
nombreUsuario: widget.password,
),
),
);
} else if (widget.password.length == 10) {
Navigator.pushReplacement(
context,
MaterialPageRoute(
builder: (context) => EstudianteHomePage(
nombreUsuario: widget.password,
),
),
);
}
);
}
}
```

```

        },
        child: Text('MENÚ PRINCIPAL'),
      ),
    ],
  ),
),
],
),
),
),
);
}
}

```

### *pagprofxadministrador.dart*

```

import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:flutter_application_notas_remoto/clases/profesor.dart';
import
'package:flutter_application_notas_remoto/login/menuadministrador.dart';
import 'package:flutter_application_notas_remoto/login/menuestudiante.dart';
import 'package:flutter_application_notas_remoto/login/menuprofesor.dart';
import 'package:flutter_application_notas_remoto/pagcursosprofesoradm.dart';
import 'package:http/http.dart' as http;

class Pxadministrador extends StatefulWidget {
final String password;

const Pxadministrador({Key? key, required this.password}) : super(key: key);

@Override
State<Pxadministrador> createState() => _PxadministradorState();
}

class _PxadministradorState extends State<Pxadministrador> {
List<Profesor>? profesores;

// Obtener datos del profesor
Future<void> getProfesor(String password) async {
final response = await http.get(
Uri.parse(
"http://192.168.42.2/ServidorNotas/Controla1.php?tag=consulta8&codAdministrador=$password"),
);
if (response.statusCode == 200) {

```

```
setState(() {
    var resp = jsonDecode(response.body);
    // Verificar si resp es una lista
    if (resp['dato'] is List){
        profesores = (resp['dato'] as List)
            .map((data) => Profesor.fromJson(data))
            .toList();
    } else {
        // Manejar el caso donde resp no es una lista (esto dependerá de cómo es la
        respuesta exacta del servidor)
        profesores = [Profesor.fromJson(resp['dato'])];
    }
});
} else {
    print('Error en la carga de datos');
}
}

@Override
void initState() {
    super.initState();
    getProfesor(widget
        .password); // Llamar a getProfesor con el código de administrador recibido
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            foregroundColor: Color.fromARGB(255, 0, 0, 0),
            backgroundColor: Color.fromARGB(255, 82, 187, 61),
            title: Text('PROFESORES'),
        ),
        body: profesores == null
            ? Center(child: CircularProgressIndicator())
            : ListView.builder(
                padding: const EdgeInsets.all(16.0),
                itemCount: profesores!.length,
                itemBuilder: (context, index) {
                    final profesor = profesores![index];
                    return Padding(
                        padding: const EdgeInsets.symmetric(vertical: 8.0),
                        child: Card(
                            elevation: 4.0,
                            child: Padding(
                                padding: const EdgeInsets.all(16.0),
                                child: Column(

```

```
crossAxisAlignment: CrossAxisAlignment.start,
children: [
  Image.network(
    "http://192.168.42.2/ServidorNotas/fotos/${profesor.codp}.jpg",
    errorBuilder: (BuildContext context,
      Object exception, StackTrace? stackTrace) {
      return Image.asset('assets/sin_foto.jpg');
    },
    width: 50,
    height: 50,
    fit: BoxFit.cover,
  ),
  Text('Código Profesor: ${profesor.codp}',
    style: TextStyle(fontSize: 16)),
  SizedBox(height: 8),
  Text('Código Especialidad: ${profesor.code}',
    style: TextStyle(fontSize: 16)),
  SizedBox(height: 8),
  Text('Nombre: ${profesor.nomp}',
    style: TextStyle(fontSize: 16)),
  SizedBox(height: 8),
  Text('Celular: ${profesor.celular}',
    style: TextStyle(fontSize: 16)),
  SizedBox(height: 8),
  Text('Dirección: ${profesor.direccp}',
    style: TextStyle(fontSize: 16)),
  SizedBox(height: 8),
  Text('Grado: ${profesor.grado}',
    style: TextStyle(fontSize: 16)),
  SizedBox(height: 8),
  Text('Años de contrato: ${profesor.anios}',
    style: TextStyle(fontSize: 16)),
  SizedBox(height: 8),
  Text('Especialidad: ${profesor.especialidad()}',
    style: TextStyle(fontSize: 16)),
  SizedBox(height: 8),
  ElevatedButton.icon(
    icon: Icon(Icons.note),
    label: Text("Ver cursos"),
    onPressed: () {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => PagCursosProfesorAdm(
            codp: profesor.codp,
            adminPassword: widget.password,
        ),
    
```



```
    },
    child: Text('MENÚ PRINCIPAL'),
),
);
}
}
```

Visor app

INICIAR SESIÓN



Usuario

Contraseña

Iniciar sesión

PANEL ADMINISTRADOR 



BIENVENIDO ADMINISTRADOR, ADM002

ADMINISTRAR PROF & ALU

[!\[\]\(7a69c04d40922ff2f2f4c2885e9b36c1\_img.jpg\) INFORMACION DE PROFESO...](#)

[!\[\]\(452ef738f5b15aff6dc23a0096440336\_img.jpg\) PROFESORES](#)



Código Profesor: PROF002

Código Especialidad: ESP002

Nombre: Ana López

Celular: 955444333

Dirección: Calle Secundaria 789, Los  
Olivos

Grado: Doctorado

Años de contrato: 6

Especialidad: Bases de Datos y  
Programación

 Ver cursos

MENÚ PRINCIPAL



## ALUMNOS POR CURSO

Investigación operativa ▾

2022101028

2022101010

2022101011

2022101012

2022101013

2022101014

2022101015



## ALUMNOS POR CURSO

Investigación operativa ▾

2022101028



≡ Gestionar notas



## Notas del Estudiante

UG

Código del Estudiante: 2022101028

Código del curso: 2005

### Curso: Investigación operativa (2005)

P1: 14.00

P2: 12.50

P3: 13.00

EF: 11.80

C1: 12.00

C2: 14.50

C3: 13.00

C4: 12.00

C5: 13.50

C6: 12.00

---

Promedio Continuas: 12.83

Promedio Curso: 12.61

MODIFICAR NOTAS

← RETORNAR



## Modificar Notas del Estudia...

Código del Estudiante: 2022101028

Código del curso: 2005

**Curso: Investigación operativa (2005)**

P1

14.0

P2

12.5

P3

13.0

EF

11.8

C1

12.0

C2

14.5

C3

13.0

**Guardar Notas**

C4

12.0

C5

13.5

C6

12.0

Promedio Continuas: 12.833333333333334

Promedio del Curso: 12.606666666666666

**Guardar Notas**

C4

12.0

C5

13.5

C6

20.0

Promedio Continuas: 14.16666666666666

Promedio del Curso: 12.87333333333335

Guardar Notas

## ← Notas del Estudiante

Código del Estudiante: 2022101028

Código del curso: 2005

**Curso: Investigación operativa (2005)**

P1: 14.00

P2: 12.50

P3: 13.00

EF: 11.80

C1: 12.00

C2: 14.50

C3: 13.00

C4: 12.00

C5: 13.50

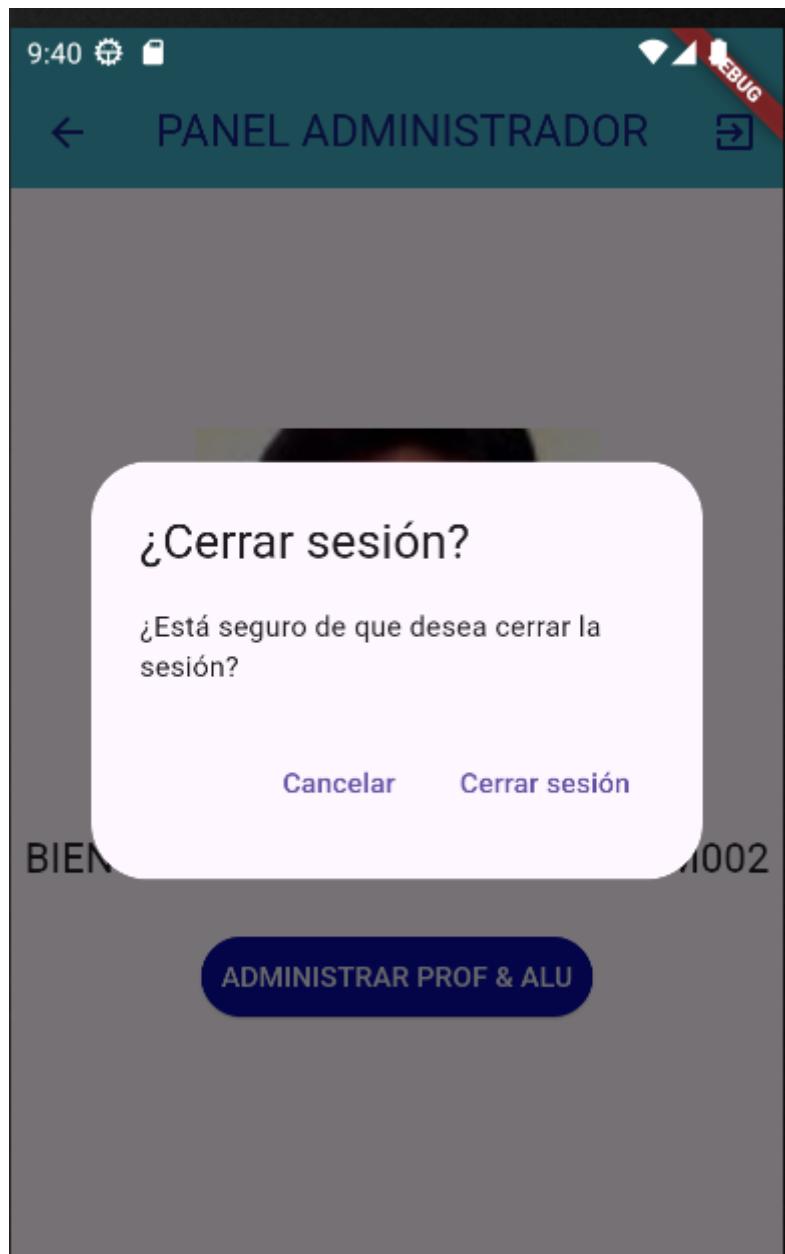
C6: 20.00

Promedio Continuas: 14.17

Promedio Curso: 12.87

MODIFICAR NOTAS

← RETORNAR



**PROFESOR**



## INICIAR SESIÓN



Usuario

@lopez

Contraseña

\*\*\*\*\*

Iniciar sesión



PANEL PROFESOR



BIENVENIDO PROFESOR, PROF002

CURSOS Y NOTAS x ALUMNO

← CURSOS Y NOTAS x ALUMNO

← ALUMNOS POR CURSO

Programación Computadoras 3 ▾

2022101028-



Villalobos Silva,  
Khleiwin

Código: 2022101028

Ver Nota

2022101010-



Gómez, Carlos

Código: 2022101010

Ver Nota

2022101011-



Martínez, Ana

Código: 2022101011

Ver Nota

2022101012-



Gomes, Anaceli

Código: 2022101012

Ver Nota

MENÚ PRINCIPAL



## Notas del Estudiante

UG

Código del Estudiante: 2022101028

Código del curso: 2000

### Curso: Programación Computadoras 3 (2000)

P1: 20.00

P2: 20.00

P3: 14.50

EF: 15.80

C1: 16.00

C2: 15.50

C3: 14.00

C4: 16.00

C5: 15.00

C6: 20.00

---

Promedio Continuas: 16.08

Promedio Curso: 16.86

MODIFICAR NOTAS

← RETORNAR

← Modificar Notas del Estudia...

Código del Estudiante: 2022101028

Código del curso: 2000

**Curso: Programación Computadoras 3  
(2000)**

P1

18

P2

18

P3

14.5

EF

15.8

C1

16.0

C2

15.5

C3

14.0

Guardar Notas

ESTUDIANTE

2022101028



## INICIAR SESIÓN



Usuario

2022101028

Contraseña

\*\*\*\*

Iniciar sesión



Panel estudiante



DEBUG



Bienvenido estudiante, 2022101028

BOLETA POR CURSO

BOLETA NOTAS

PAGOS

PLAN ACADÉMICO



## Cursos Estudiante

UG

### 2000 - Programación Computadoras 3 ▾



user

**Curso:Programación Computadoras 3 - 2000**

**Prof: Ana López**

**EC[20%] P1[10%] P2[20%] P3[20%] Ef[30%]**

16.08    18.00    18.00    14.50    14.50

**C1              C2              C3              C4**

16.00        15.50        14.00        16.00

**C5              C6**

15.00                          20.00

**MENÚ PRINCIPAL**

**Curso: Programación Computadoras 3 - 2000**

**Prof: Ana López**

**EC[20%] P1[10%] P2[20%] P3[20%] Ef[30%]**

16.08	18.00	18.00	14.50	14.50
-------	-------	-------	-------	-------

<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>
-----------	-----------	-----------	-----------

16.00	15.50	14.00	16.00
-------	-------	-------	-------

<b>C5</b>	<b>C6</b>
-----------	-----------

15.00	20.00
-------	-------

Promedio: 16.26

**VER BOLETA NOTAS**

**Curso: Programación  
Computadoras 3 COD: 2000**

**Prof: Ana López**

**EC[20%] P1[10%] P2[20%] P3[20%] Ef[30%]**

16.08	18.00	18.00	14.50	15.80
-------	-------	-------	-------	-------

<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>
-----------	-----------	-----------	-----------

16.00	15.50	14.00	16.00
-------	-------	-------	-------

<b>C5</b>	<b>C6</b>
-----------	-----------

15.00	20.00
-------	-------

**MENÚ PRINCIPAL**

**VER BOLETA NOTAS**

**Curso: Microprocesadores COD:  
2002**

**Prof: Adrian Piedra**

**EC[20%] P1[10%] P2[20%] P3[20%] Ef[30%]**

15.17    15.50    16.00    14.50    15.80

C1	C2	C3	C4
16.00	15.50	14.00	16.00
C5	C6		
15.00	14.50		

Promedio: 15.42

**MENÚ PRINCIPAL**



## CONSULTA PAGOS

DEBUG

### Cuota: 1

Monto: 550.00

Descuento considerado: 55.00

Fecha vencimiento: 2024-03-15 00:00:00

### Cuota: 2

Monto: 550.00

Descuento considerado: 55.00

Fecha vencimiento: 2024-04-15 00:00:00

### Cuota: 3

Monto: 567.00

Descuento considerado: 0.00

Fecha vencimiento: 2024-05-15 00:00:00

**MENÚ PRINCIPAL**

### Cuota: 5

Monto: 550.00

Descuento considerado: 0.00

Fecha vencimiento: 2024-07-10 00:00:00

Pago total: 2729.00

**MENÚ PRINCIPAL**



9:47

DEBUG

## ← Plan de Estudios

1



**Código: 150286**

**Asignatura: ACTIVIDADES I**

Tipo: O

HT: 0

HP: 2

CRD: 3

Prerrequisitos: 1

**Código: 170001**

**Asignatura: ANTROPOLOGIA RELIGIOSA**

Tipo: O

HT: 4

HP: 0

CRD: 3

Prerrequisitos: 4

## Plan de Estudios

CRD: 3

Prerrequisitos: (III) 132105

Código: 200265

Asignatura: DESARROLLO BASADO EN  
PLATAFORMAS

Tipo: E

HT: 2

HP: 2

CRD: 3

Prerrequisitos: (III) 200261

### INDICACIONES

- TIPO: Asignatura Obligatoria (O) o Electiva (E).
- HT: HORAS DE TEORIA requeridas por la asignatura.
- HP: HORAS DE PRACTICA requeridas por la asignatura.
- CRD: CRÉDITOS correspondientes a la asignatura.

[MENÚ PRINCIPAL](#)

5

**Código:** 200242

**Asignatura:** BASE DE DATOS 2

Tipo: O

HT: 2

HP: 2

CRD: 3

Prerrequisitos: (IV) 200223

**Código:** 200268

**Asignatura:** FUNDAMENTOS DE SISTEMAS  
OPERATIVOS

Tipo: O

HT: 2

HP: 2

CRD: 3

Prerrequisitos: (IV) 200223

## Plan de Estudios

CRD: 3

Prerrequisitos: (IV) 200294

**Código: 200266**

**Asignatura: DISEÑO DE ALGORITMOS**

Tipo: E

HT: 2

HP: 2

CRD: 3

Prerrequisitos: (IV) 200294

### **INDICACIONES**

- **TIPO:** Asignatura Obligatoria (O) o Electiva (E).
- **HT:** HORAS DE TEORIA requeridas por la asignatura.
- **HP:** HORAS DE PRACTICA requeridas por la asignatura.
- **CRD:** CRÉDITOS correspondientes a la asignatura.

**MENÚ PRINCIPAL**

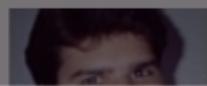
9:49



DEBUG



Panel estudiante



¿Cerrar sesión?

B

¿Está seguro de que desea cerrar la sesión?

8

Cancelar

Cerrar sesión

PAGOS

PLAN ACADÉMICO

← Notas del Estudiante

BUG

Código del Estudiante: 2022101028

Código del curso: 2000

**Curso: Programación Computadoras 3  
(2000)**

P1: 18.00

P2: 18.00

P3: 14.50

EF: 15.80

C1: 16.00

C2: 15.50

C3: 14.00

C4: 16.00

C5: 15.00

C6: 20.00

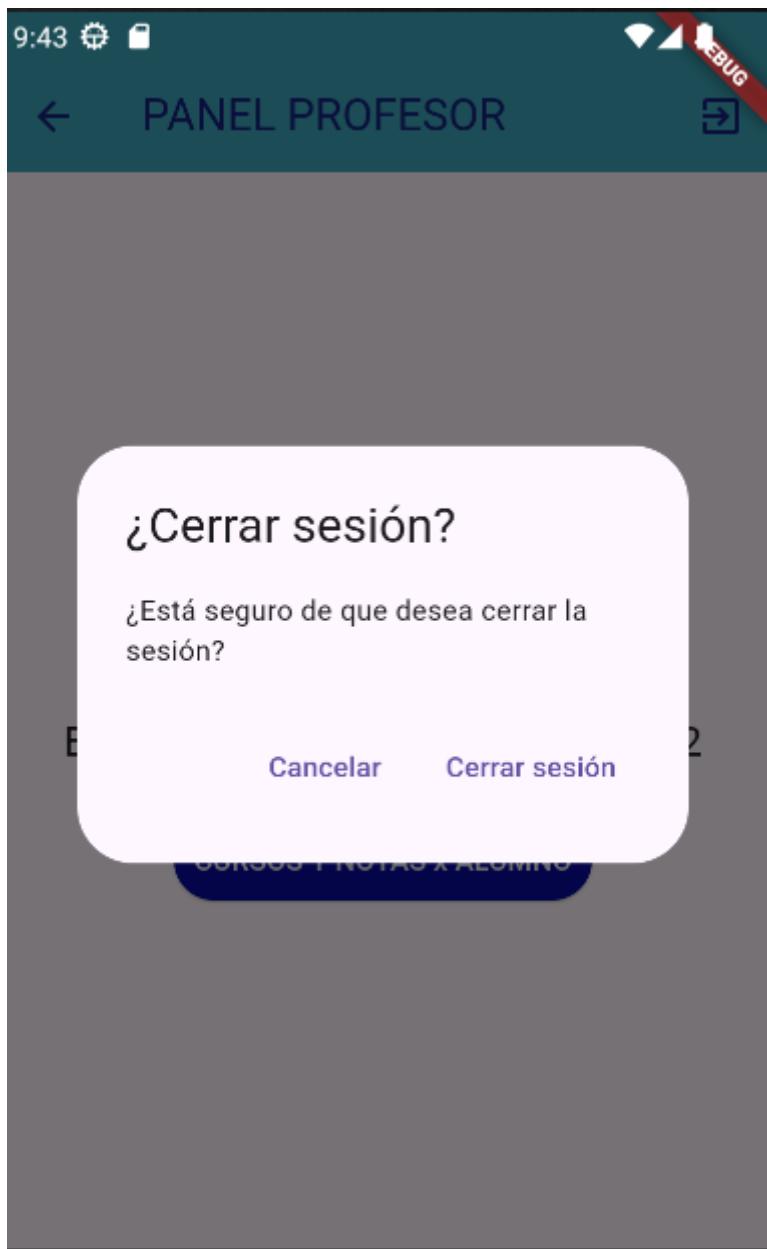
---

Promedio Continuas: 16.08

Promedio Curso: 16.26

MODIFICAR NOTAS

← RETORNAR



## Sección 1: Procedimiento documentado de la instalación de Jenkins:

### 1. Requisitos previos

Antes de comenzar con la instalación de Jenkins en Windows, asegúrate de tener los siguientes requisitos:

**Java:** Jenkins está basado en Java, por lo tanto, es necesario tener una versión compatible instalada, recomendable el jdk17 o 21.

Siguiente enlace jdk21 de java:

1

<https://www.oracle.com/java/technologies/javase/jdk21-archive-downloads.html>

**Herramientas de red:** Asegúrate de tener una conexión a Internet estable para descargar Jenkins y sus dependencias.

## 2. Pasos de instalaciones

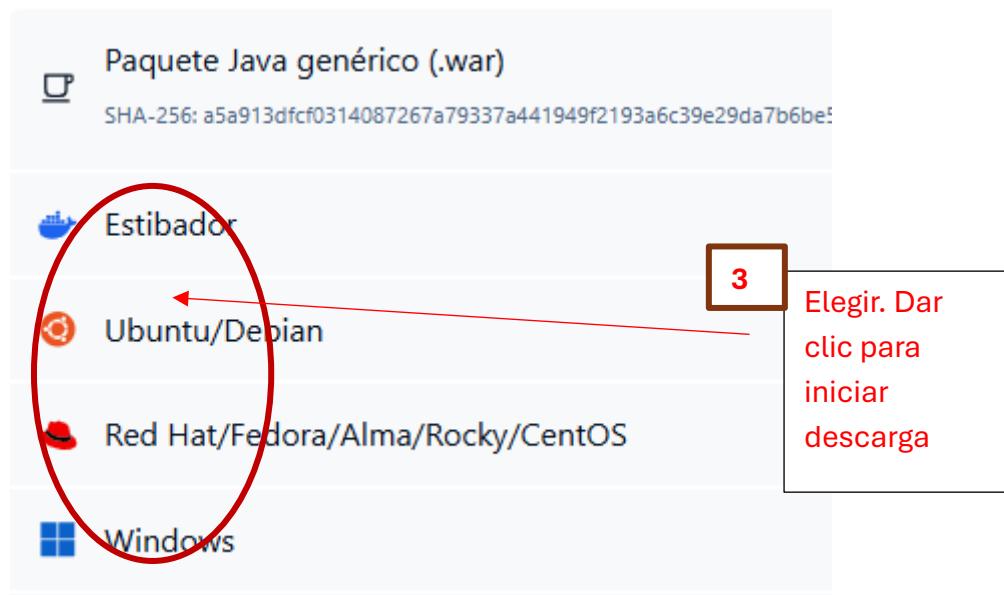
### 2.1 Descargar Jenkins

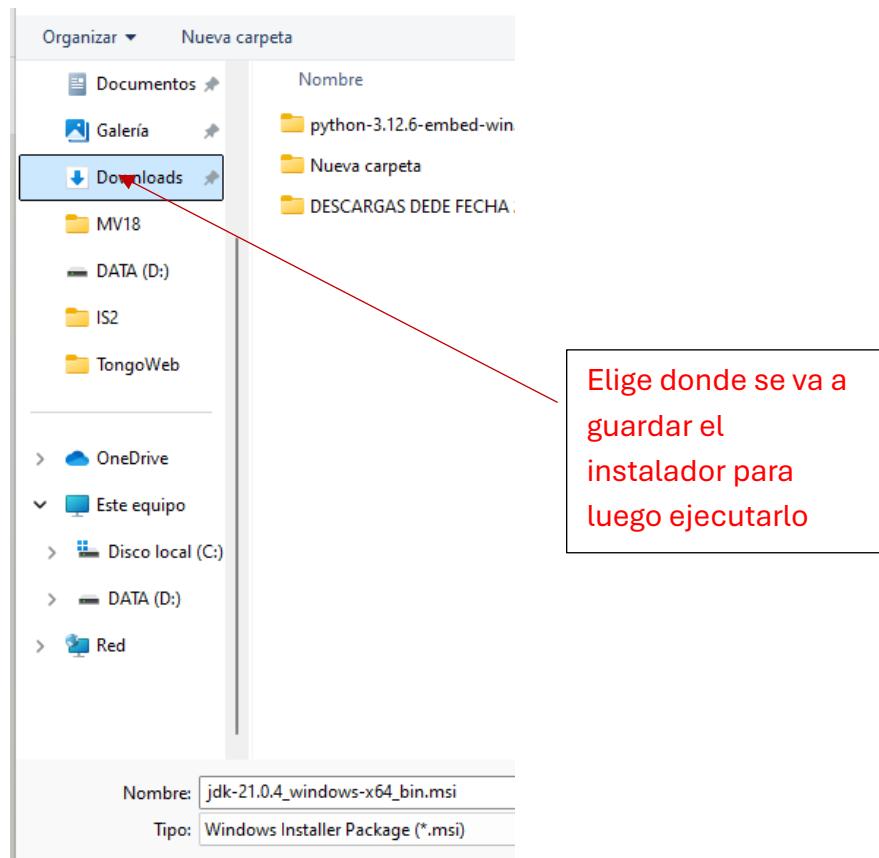
Dirígete al sitio web oficial de Jenkins en

<https://www.jenkins.io/download>

2

Descarga el instalador adecuado para el SO correspondiente, generalmente el archivo con extensión .war o el archivo .msi.

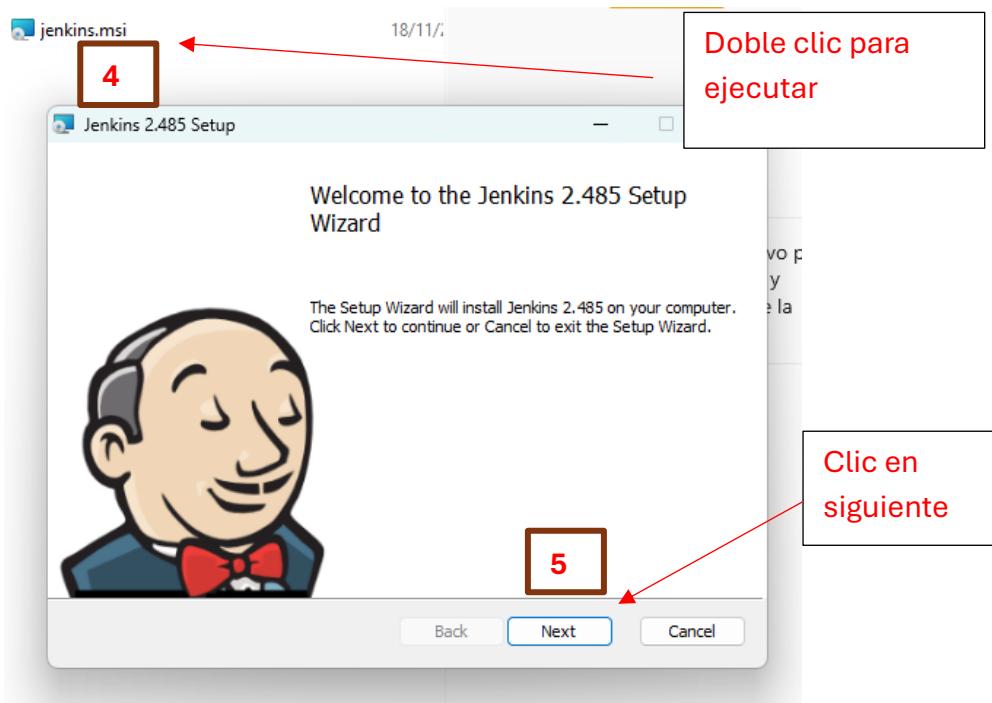




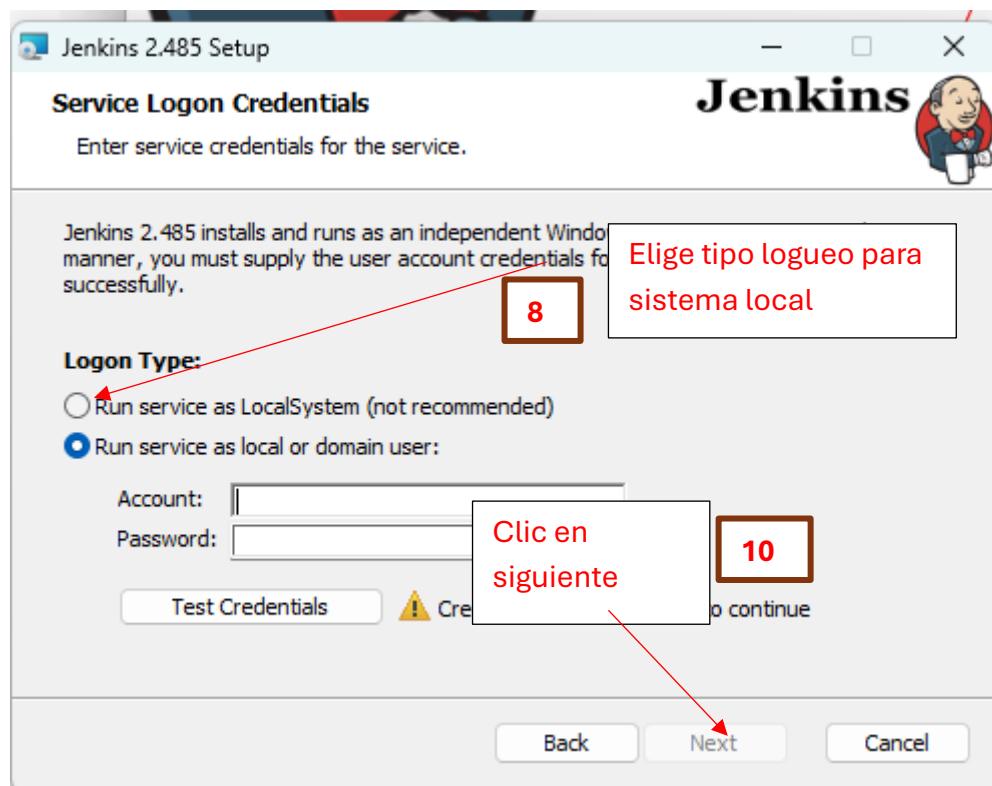
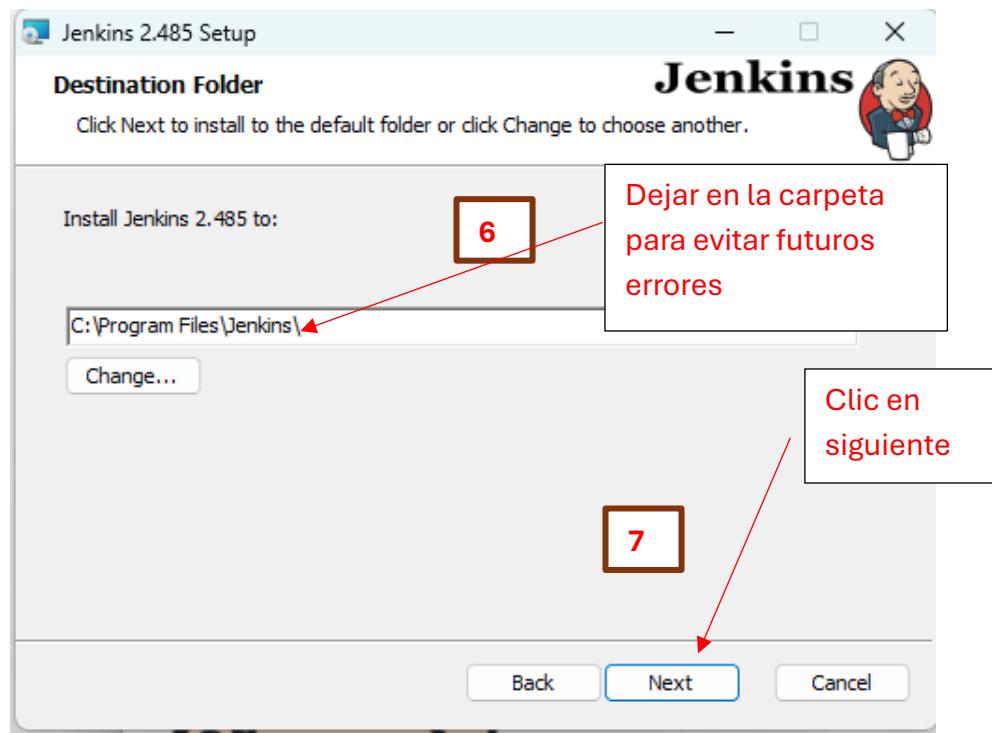
## 2.2. Instalar Jenkins:

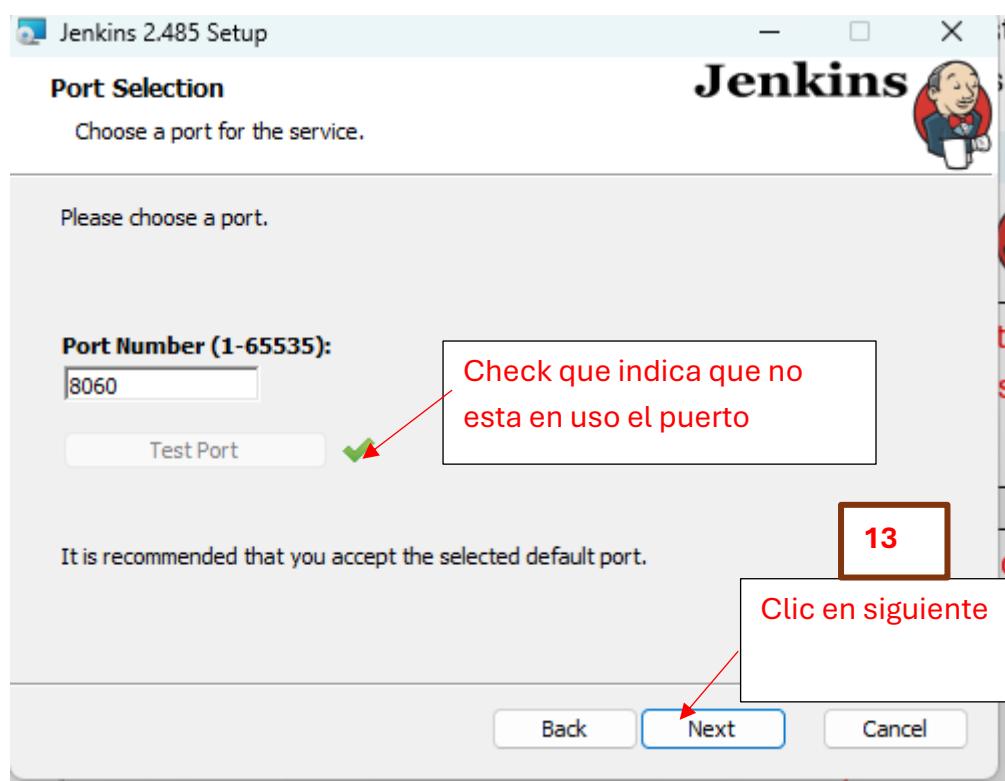
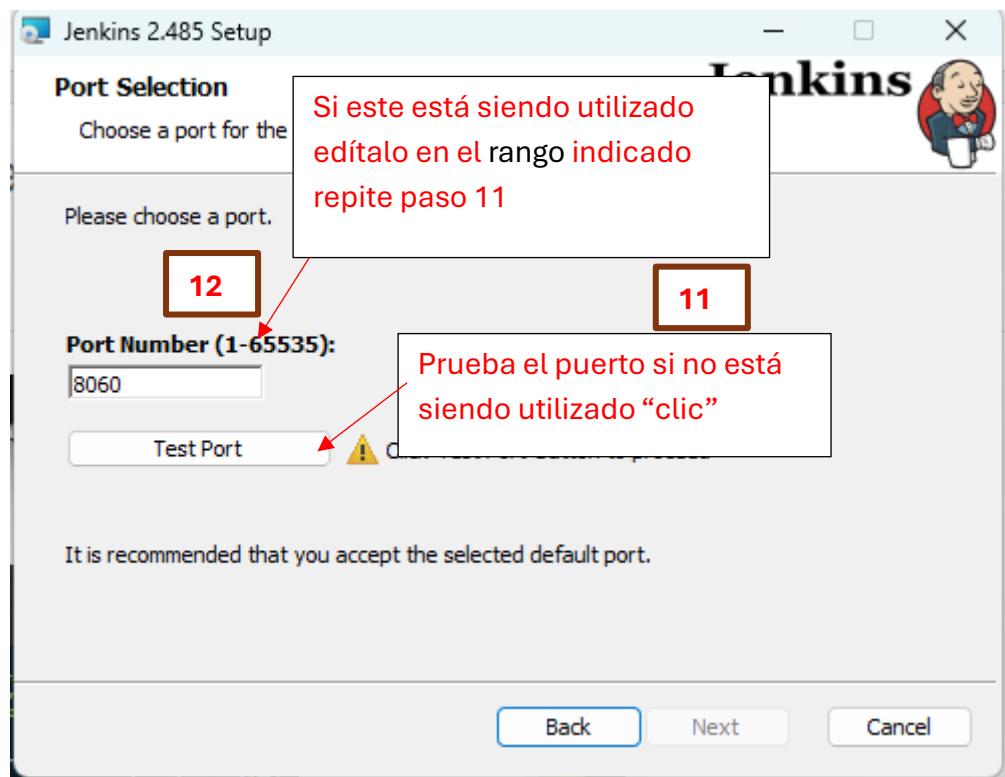
Si has descargado el instalador **.msi**, sigue estos pasos:

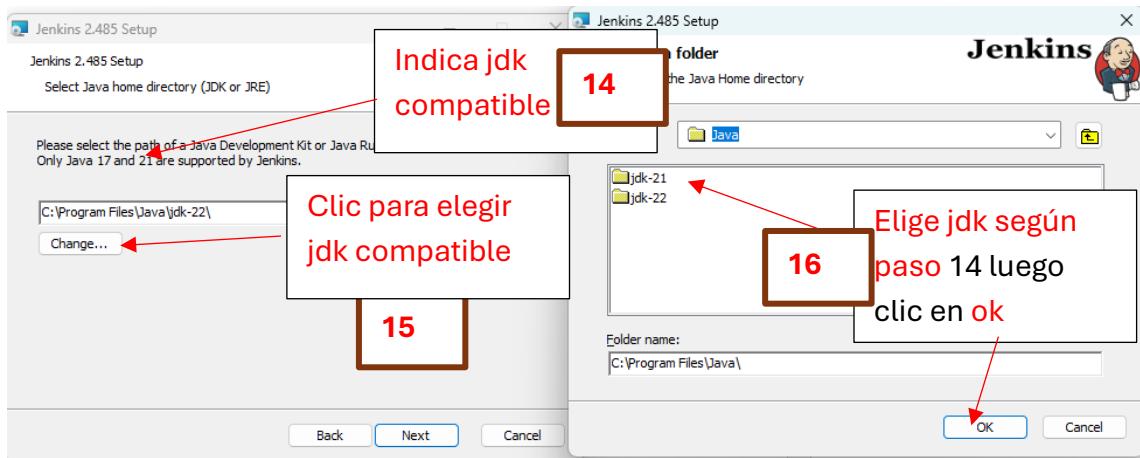
- 1) Accede a la carpeta donde guardaste el archivo del instalador.
- 2) Ejecuta el archivo **.msi**.



- 3) Acepta los términos de licencia y selecciona la carpeta de destino para la instalación (por defecto C:\Program Files (x86) \Jenkins).



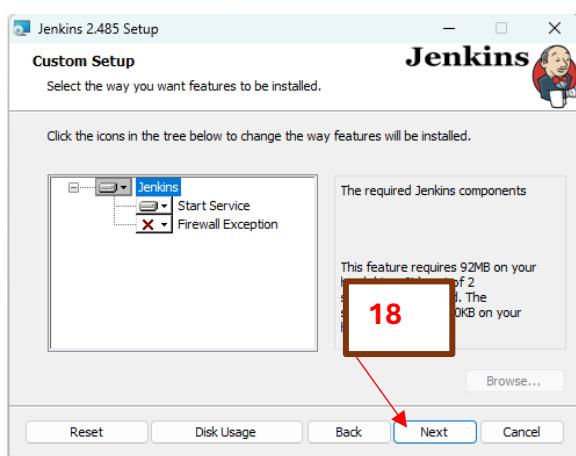




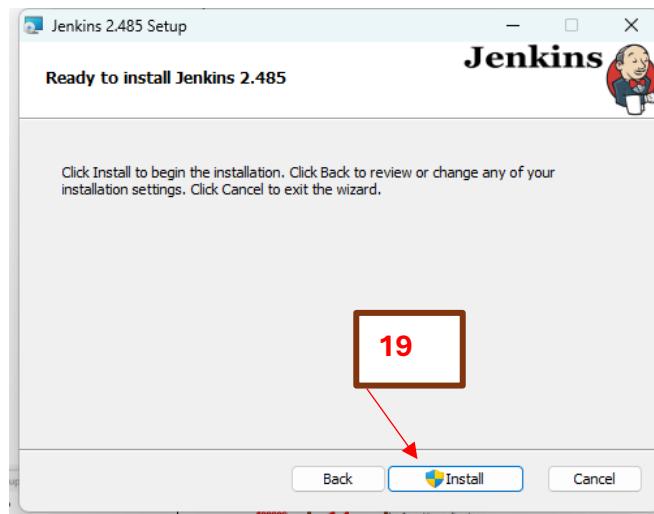
- 4) Una vez elegido el JDK compatible clic en siguiente. O si no cuentas con un JDK compatible clic en el enlace del **paso 1 y continuar**:



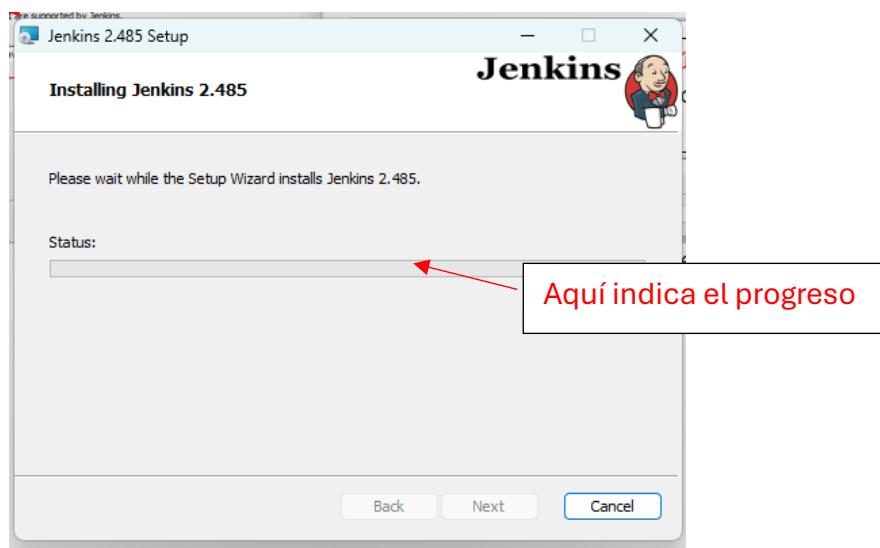
- 5) Clic en siguiente



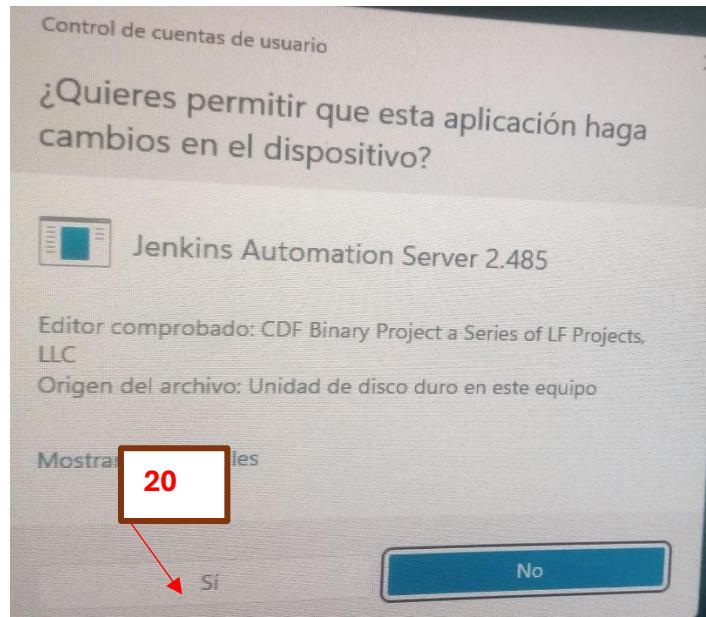
6) Clic en install:



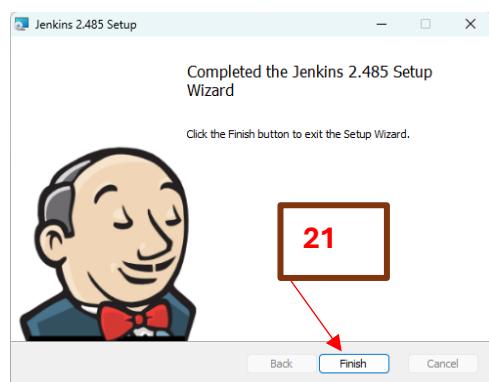
7) Esperar que el proceso se complete:



- 8) Una vez terminado te preguntara si quieres permitir que la aplicación haga cambios en el dispositivo dar que sí.



- 9) Dar clic en terminar:

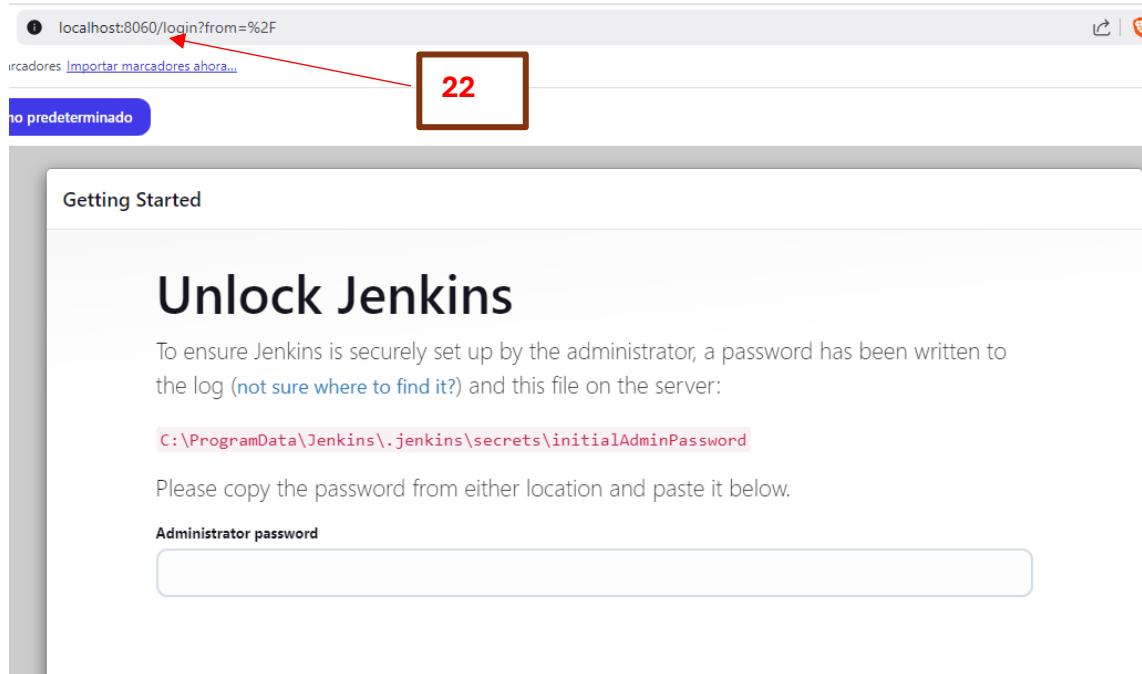


- 10) Asegúrate de seleccionar la opción para iniciar Jenkins como un servicio de Windows durante la instalación.

### 2.3. Verificar la instalación de Jenkins:

Después de completar la instalación, abre un navegador web y accede a la URL <http://localhost:8060>.

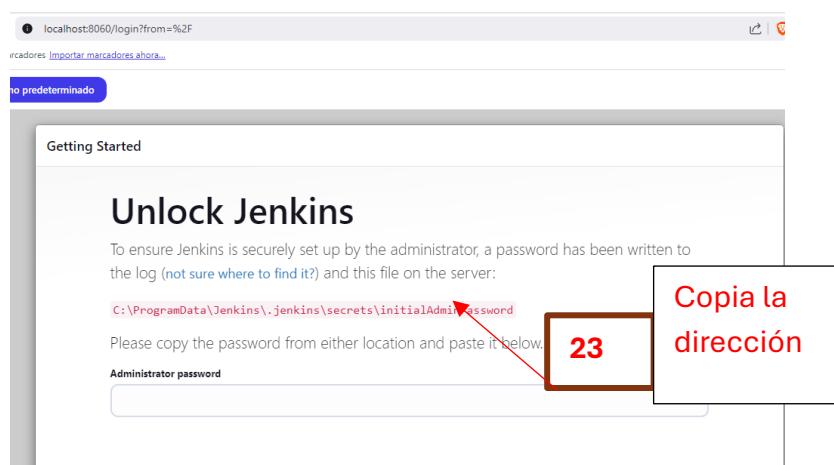
**Si todo ha ido bien, verás la página de inicio de Jenkins.**



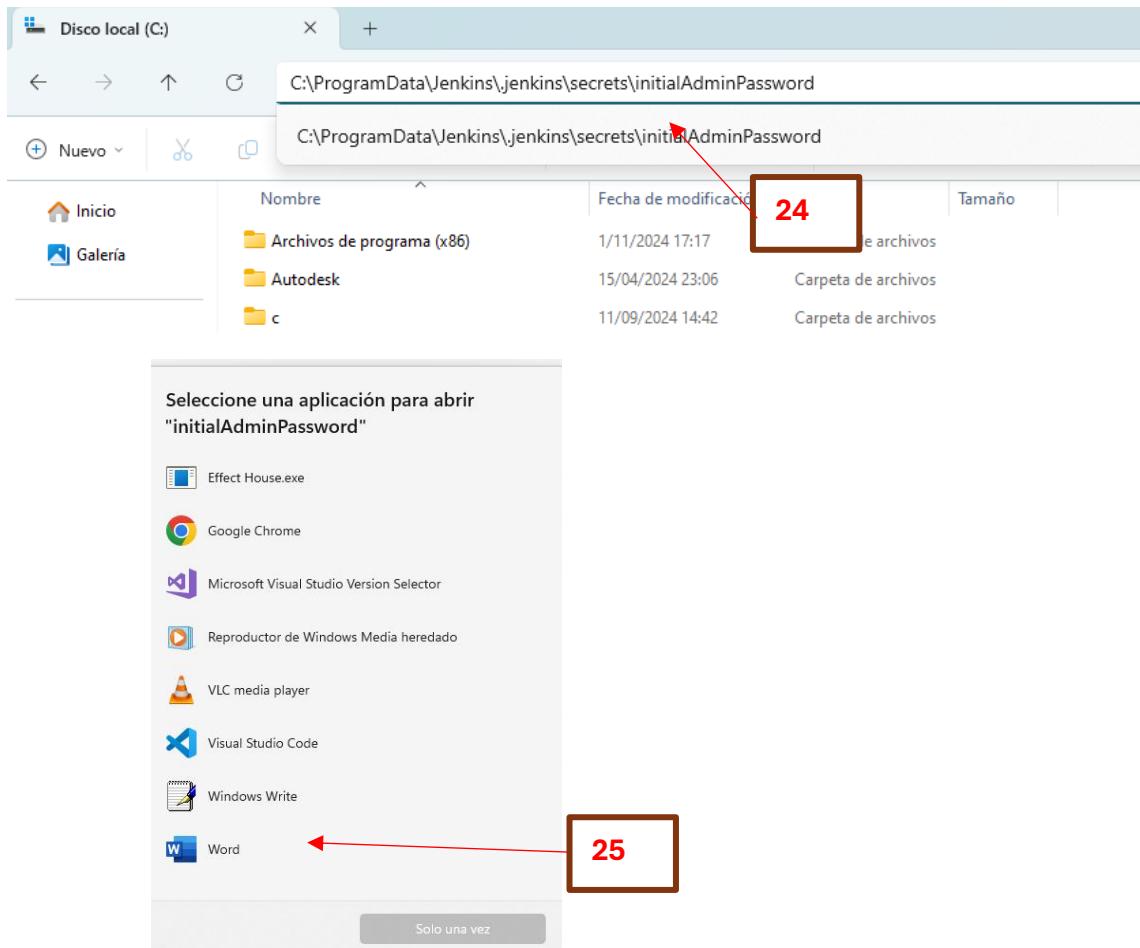
## 2.4. Desbloquear Jenkins

- Durante la primera ejecución, Jenkins generará una clave de desbloqueo. Esta clave se encuentra en el archivo secrets/initialAdminPassword dentro del directorio de instalación de Jenkins (C:\Program Files (x86)\Jenkins\ o el directorio donde lo hayas instalado).

C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword



- Ejecuta la dirección y elige recomendable **block de notas** para abrir el archivo



Copia la clave y pégala en el campo correspondiente en la página web de Jenkins luego clic en continuar.

## Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (*not sure where to find it?*) and this file on the server:

C:\ProgramData\Jenkins\jenkins\secrets\initialAdminPassword

Please copy the password from either location and paste it below.

Administrator password

.....

26

27

Continue

## 2.5. Configurar usuario administrador

- Después de desbloquear Jenkins, se te pedirá que crees un usuario administrador. Completa los campos requeridos y luego clic en **guardar y continuar** **EJEMPLO:**

### Create First Admin User

Usuario  
khleiwin

Contraseña  
.....

Confirma la contraseña  
.....

Nombre completo  
KHLEIWIN VILLALOBOS SILVA

Dirección de email  
villaloboskhleiwin@gmail.com

Skip and continue as admin **Save and Continue**

Confirmar la instancia de configuración clic en: **save and finish**

### Instance Configuration

Jenkins URL: http://localhost:8060/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD\_URL environment variable provided to build steps.  
The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Not now **Save and Finish**

## 2.6. Instalar plugins recomendados

- Durante la configuración inicial, Jenkins te ofrecerá instalar los plugins recomendados. **Haz clic en "Install suggested plugins".** Esto instalará los plugins necesarios para el funcionamiento básico de Jenkins.

# Bienvenido a Jenkins

Plugins extend Jenkins with additional features to support many different needs.

29

## Install suggested plugins

Install plugins the Jenkins community finds most useful.

## Select plugins to install

Select and install plugins most suitable for your needs.

- Finalmente, ya podemos utilizar Jenkins:

## Getting Started

30

# Jenkins is Ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

## 2.7. Registro e iniciar sesión en NGROK

El objetivo de Ngrok es permitir exponer un servidor local a Internet de manera rápida y sencilla. Ya que nuestro servidor de Jenkins es local, paso 8 de instalación de Jenkins, Ngrok creará un túnel seguro para un acceso remoto. Esto nos servirá para configurar la integración continua con nuestro proyecto de GitHub por medio de los WebHooks.

- Lo primero que se debe hacer es dirigirnos a la página oficial de Ngrok con el siguiente enlace: <https://ngrok.com>. Es necesario registrarse si no tenemos cuenta y luego iniciar sesión.



- El proceso de registrarse puedes elegir entre rellenar los campos o elegir una opción de registrarse con GITHUB o Google. En este caso elige iniciar con GitHub.

### Únete

Nombre

Correo electrónico

Contraseña  
 ⚡

Acepto los [términos de servicio](#) y la política de [privacidad](#)

**Únete**

o

 [Regístrate con GitHub](#)

 [Regístrate en Google](#)

¿Ya tienes una cuenta?  
[¡Inicia sesión aquí!](#)



- Se pasará a aceptar la autorización de uso:

 ngrok.com por [ngrok-private](#)  
quiere acceder a su cuenta de [cuenta](#)

 **Datos personales del usuario**  
Direcciones de correo electrónico (solo lectura)

**Autorizar ngrok-private**

Al autorizar, se redirigirá a  
<https://dashboard.ngrok.com>

 No es propiedad ni  
está operado por  
GitHub

 Creado Hace 12 años

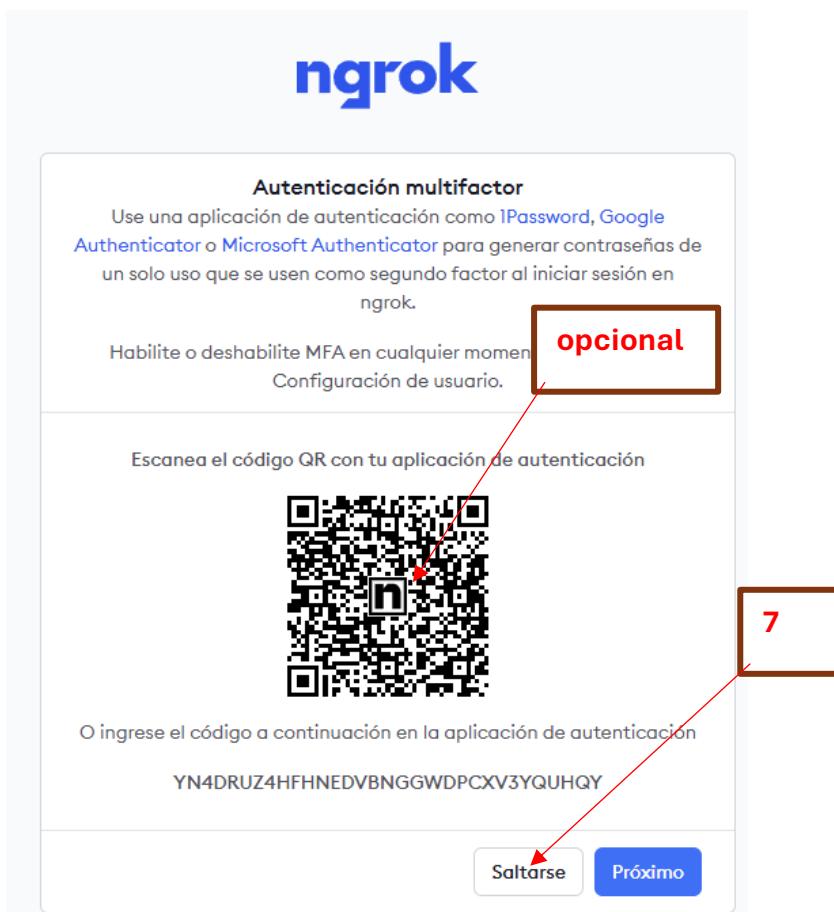
 Más de 1K  
Usuarios de GitHub



- Confirmar la aceptación de términos y luego clic en Crear una cuenta:



- Elegir entre escanear el QR con una aplicación de autenticación o simplemente clic en saltarse:





#### Autenticación multifactor

La autenticación multifactor (MFA) se puede habilitar en cualquier momento desde la página [Configuración de usuario](#)

Entiendo

8

- Elegir según sea para lo que usa y luego clic en continuar:



#### ¡Bienvenidos a ngrok!

¿Cómo te describirías a tí mismo?

Estudiante / Aficionado (Construir por diversión)

9

¿Para qué le interesa usar ngrok?

- Proxy con reconocimiento de identidad
- Puerta de enlace de API
- Compartir aplicaciones locales sin implementarlas
- Kubernetes Ingress
- Conectividad de dispositivos IoT
- Prueba de webhooks en local
- API de puerta de enlace de Kubernetes
- Conexión a API o bases de datos en las redes de sus clientes
- Mi caso de uso no está aquí: respuesta personalizada

10

Estoy construyendo algo más...

¿Está utilizando ngrok para

Producción

Desarrollo

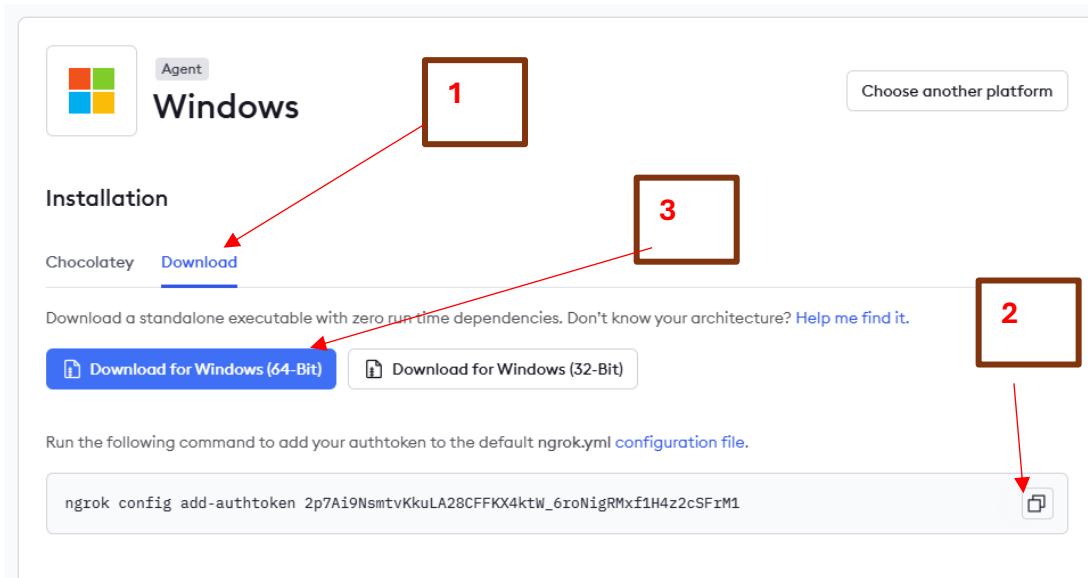
11

Continuar

12

## 2.8. Descargar NGROK:

- Una vez registrada la cuenta e iniciado la sesión, nos redirige a la ventana principal, en donde debemos seleccionar la pestaña '**Download**'. La página detecta automáticamente el sistema operativo de nuestra PC. Antes de bajar al programa, copiamos el comando para crear el ngrok.yml. Descargamos el ejecutable y guardamos en una carpeta que se elija en la computadora.



- Una vez guardado y descomprimido el archivo pasamos a ejecutar el instalador:

Nueva carpeta

	Nombre	Fecha de modificación	Tipo	Tamaño
Documentos	CVS KHLEIWIN	10/09/2024 15:30	Carpeta de archivos	
Galería	html-docs	25/10/2024 15:32	Carpeta de archivos	
Downloads	IS2	20/11/2024 08:01	Carpeta de archivos	
MV18	KHLEIWIN 2024-I (V)	16/11/2024 18:15	Carpeta de archivos	
DATA (D:)	MUSICA NUEVA	18/11/2024 17:08	Carpeta de archivos	
IS2	Otros archivos	16/11/2024 20:58	Carpeta de archivos	
TongoWeb	Otros archivos d	28/10/2024 11:49	Carpeta de archivos	
	parcial 2 p1 y p2	19/10/2024 07:38	Carpeta de archivos	
OneDrive	Programas Python	1/11/2024 19:20	Carpeta de archivos	
Este equipo	PYTHON	1/11/2024 17:29	Carpeta de archivos	
Disco local (C:)	RECICLAJE TEMPORAL	30/08/2024 22:32	Carpeta de archivos	
DATA (D:)	TeraBoxDownload	4/10/2024 16:52	Carpeta de archivos	
Red	TongoM	13/11/2024 08:41	Carpeta de archivos	
	TongoWeb	29/10/2024 15:46	Carpeta de archivos	
	IS2.zip	16/10/2024 17:28	Carpeta comprimi...	24,143 KB
	PYTHON.zip	1/11/2024 17:27	Carpeta comprimi...	16 KB

Nombre: ngrok-v3-stable-windows-amd64.zip  
Tipo: Archivo WinRAR ZIP (\*.zip)

- Doble clic en el archivo de ejecución para proceder a la descarga:

Nombre	Fecha de modificación	Tipo	Tamaño
ngrok.exe	20/11/2024 08:45	Aplicación	47,018 KB

- Muestra una interfaz en el ejecutor de comandos donde debemos pegar el comando del **paso 3** y luego enter:

```

ngrok - tunnel local ports to public URLs and inspect traffic

USAGE:
  ngrok [command] [flags]

AUTHOR:
  ngrok - <support@ngrok.com>

COMMANDS:
  config      update or migrate ngrok's configuration file
  http        start an HTTP tunnel
  tcp         start a TCP tunnel
  tunnel      start a tunnel for use with a tunnel-group backend

EXAMPLES:
  ngrok http 80                                # secure public URL for port 80 web server
  ngrok http --url baz.ngrok.dev 8080           # port 8080 available at baz.ngrok.dev
  ngrok tcp 22                                  # tunnel arbitrary TCP traffic to port 22
  ngrok http 80 --oauth=google --oauth-allow-email=foo@foo.com # secure your app with oauth

Paid Features:
  ngrok http 80 --url mydomain.com              # run ngrok with your own custom domain
  ngrok http 80 --cidr-allow 2600:8c00::a03c:91ee:fe69:9695/32 # run ngrok with IP policy restrictions
  Upgrade your account at https://dashboard.ngrok.com/billing/subscription to access paid features

Upgrade your account at https://dashboard.ngrok.com/billing/subscription to access paid features

Flags:
  -h, --help     help for ngrok

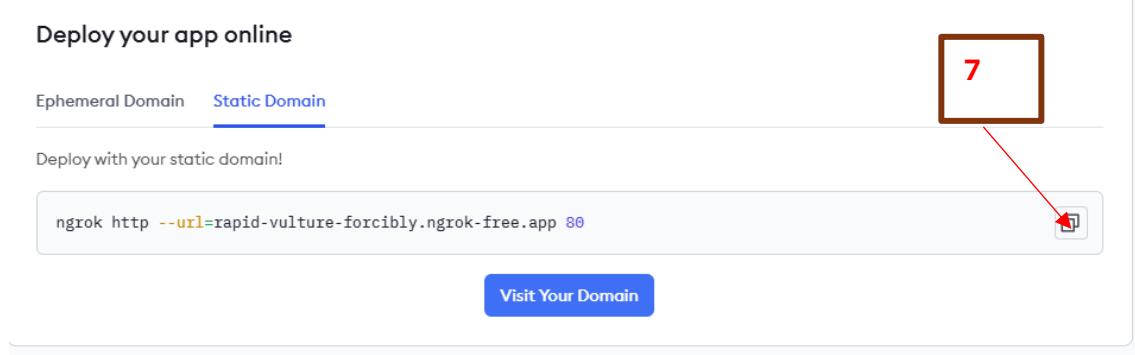
Use "ngrok [command] --help" for more information about a command.

ngrok is a command line application, try typing 'ngrok.exe http 80'
at this terminal prompt to expose port 80.
D:\IS2\ngrok-v3-stable-windows-amd64>ngrok config add-authtoken 2p7Ai9NsmtvKkuLA28CFFKX4ktW_6roNigRMxf1H4z2cSFmI
Authtoken saved to configuration file: C:\Users\villa\AppData\Local/ngrok/ngrok.yml

D:\IS2\ngrok-v3-stable-windows-amd64>_

```

- Nos dirigimos nuevamente a la ventana principal de Ngrok de nuestro navegador. En la sección ‘Deploy your app online’ seleccionamos la pestaña de “Static Domain” y copiamos el comando y lo pegamos en el ejecutor de comandos del paso 6, sin antes cambiar el último número al correspondiente puerto de Jenkins (8060). Esto nos permitirá mantener una única URL para acceder a nuestro servidor de Jenkins de manera remota, ya que si utilizamos el comando que nos ofrece la página, cada vez que apaguemos nuestro servidor (nuestra PC) debemos crear una nueva URL, impidiendo la integración continua que necesitamos establecer.



- El paso 7 y ejecución de ese comando hay que repetirlos siempre que prendemos nuevamente nuestro servidor (nuestra PC en donde está instalado el Jenkins). Ahora, la consola nos ofrecerá la URL para



acceder a Jenkins de manera remota. Copiamos, ya que nos ayudará más adelante.

```
ngrok
Help shape K8s Bindings https://ngrok.com/new-features-update?ref=k8s

Session Status          online
Account                 KhleiwinMPcv (Plan: Free)
Version                3.18.4
Region                 South America (sa)
Latency                102ms
Web Interface          http://127.0.0.1:4040
Forwarding             https://rapid-vulture-forcibly.ngrok-free.app -> http://localhost:8060

Connections            ttl     opn      rt1     rt5      p50      p90
                        0       0       0.00    0.00    0.00    0.00
```

## 2.9. Instalación del GITHUB

- La instalación de Git debe hacerse tanto en el servidor de Jenkins como en el PC donde tengamos el proyecto que deseamos subir al repositorio. La instalación va a depender del tipo de sistema operativo que se tenga, pero en este caso para Windows, se puede descargar con el siguiente enlace: <https://git-scm.com>. Descargamos el ejecutable y seguimos los pasos para la correcta instalación del GitHub.

The screenshot shows the GitHub homepage with the title "git --control-de-versiones-rapidas". Below the title, there is a brief description of Git as a distributed version control system. To the right, there is a diagram illustrating a branching workflow with four repositories connected by arrows. A red box labeled "1" points to a monitor displaying the GitHub Windows desktop application interface, which shows the "Última versión de origen 2.47.0" and a "Descargar para Windows" button. A red arrow points from the "Descargar para Windows" button to the "Windows" link in the footer. The footer also includes links for "Documentación", "Comunidad", "Interfaces gráficas de usuario de Windows", "Bolas de alquitrán", "Construcción para Mac", and "Código fuente".

- Elige según tu preferencia:

# Descargar para Windows

Haga clic aquí para descargar la última versión (2.47.0(2)) de 64 bits de Git para Windows. Esta es la compilación mantenida más reciente. Fue lanzado hace 29 días, el 20 de octubre de 2023.

2

## Otras descargas de Git para Windows

Instalador independiente

[Git de 32 bits para la instalación de Windows.](#)

[Git de 64 bits para la instalación de Windows.](#)

Portátil ("edición de memoria USB")

[Git de 32 bits para Windows Portable.](#)

[Git de 64 bits para Windows Portable.](#)

## Uso de la herramienta winget

Instale la herramienta winget si aún no la tiene, luego escriba este comando en el símbolo del sistema o Powershell.

```
winget install --id Git.Git -e --source winget
```

La versión actual del código fuente es la versión 2.47.0. Si desea la versión más reciente, puede compilarla a partir del código fuente.

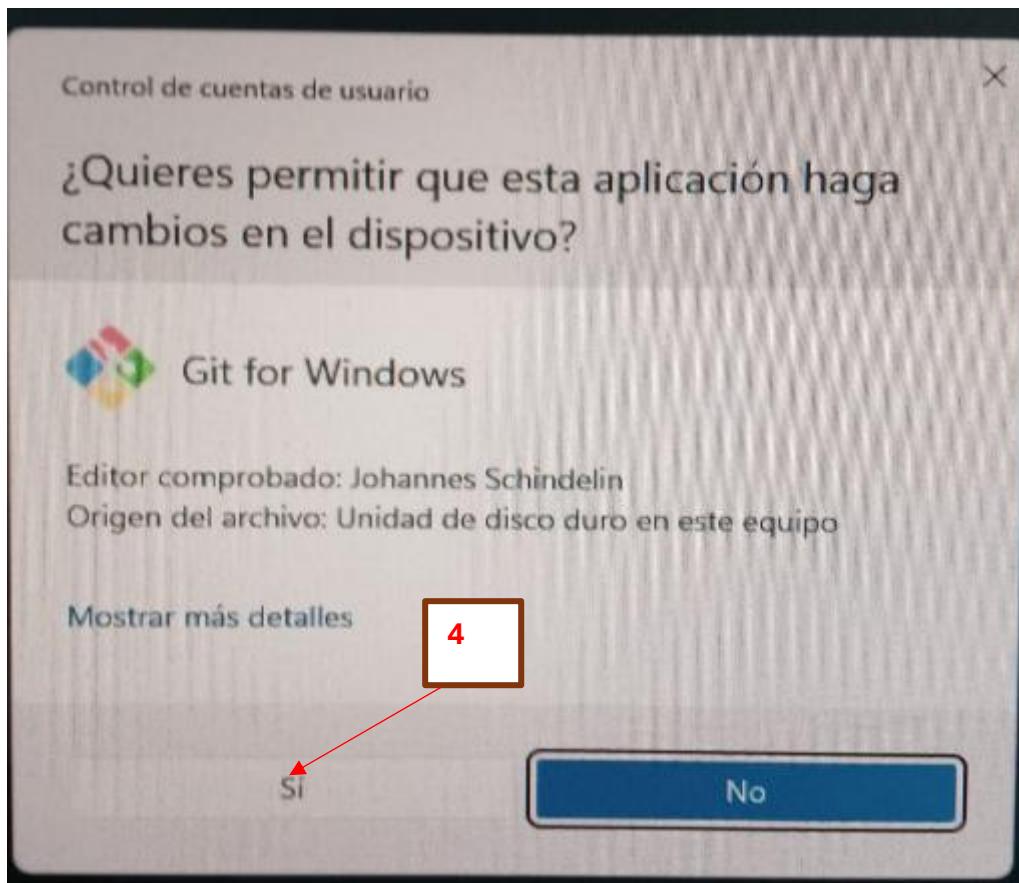
## ¿Y ahora qué?

Ahora que has descargado Git, es hora de empezar a usarlo.

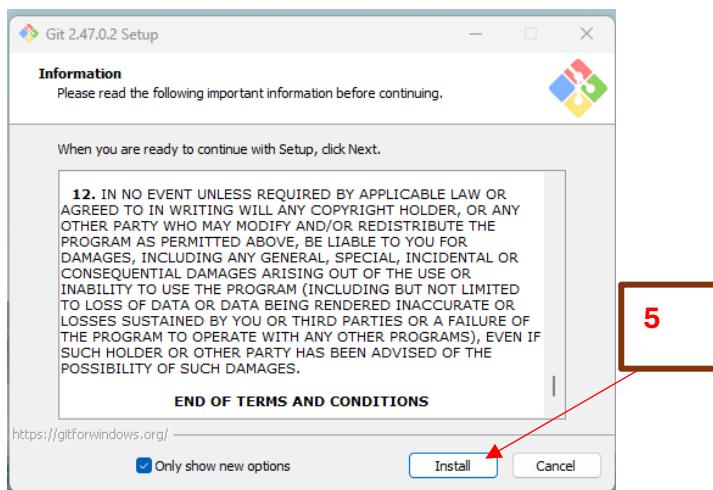
- Abrir archivo y ejecutar:



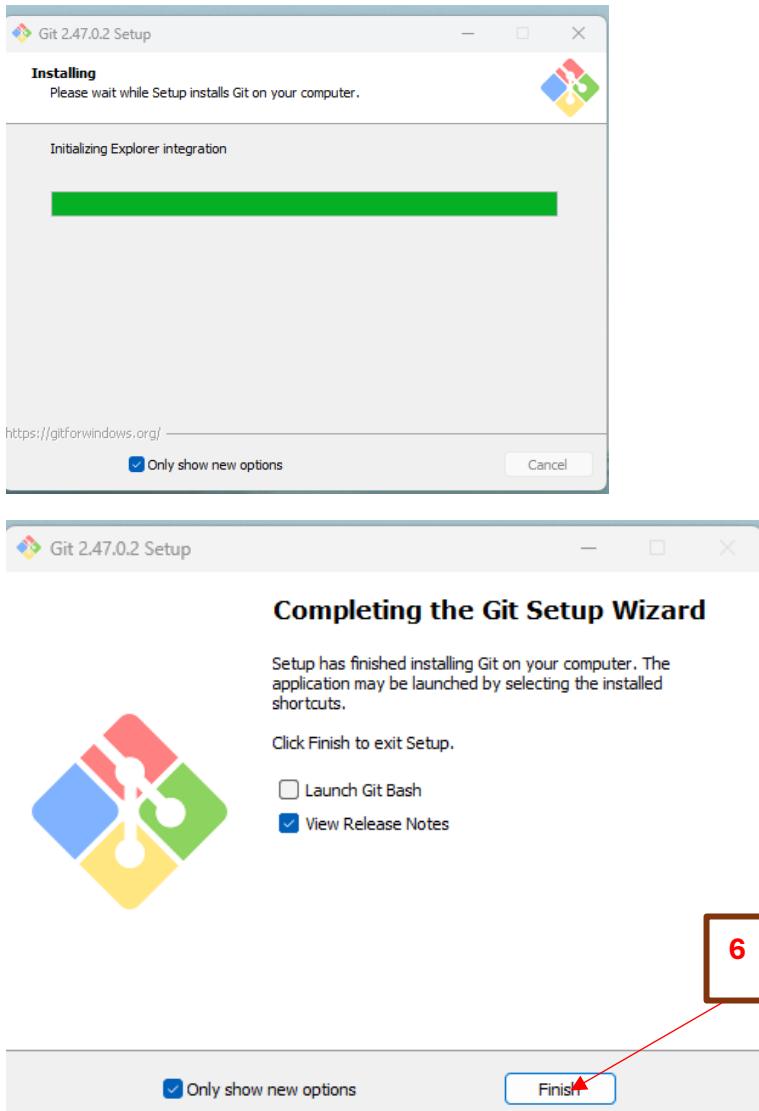
- Clic en si para permitir instalación:



- Clic en Install:



- Esperar que se complete la instalación y clic en **Finish**:



## 2.10. Instalar Visual Studio Code:

- Para descargar VSCode nos tenemos que dirigir al siguiente enlace: <https://code.visualstudio.com>. Sigue los siguientes pasos para la instalación. Debes descargar el instalador y guardarlo en la carpeta que creas correspondiente. Accedemos a él mediante el navegador preferido.

**Codifique más rápido con IA**

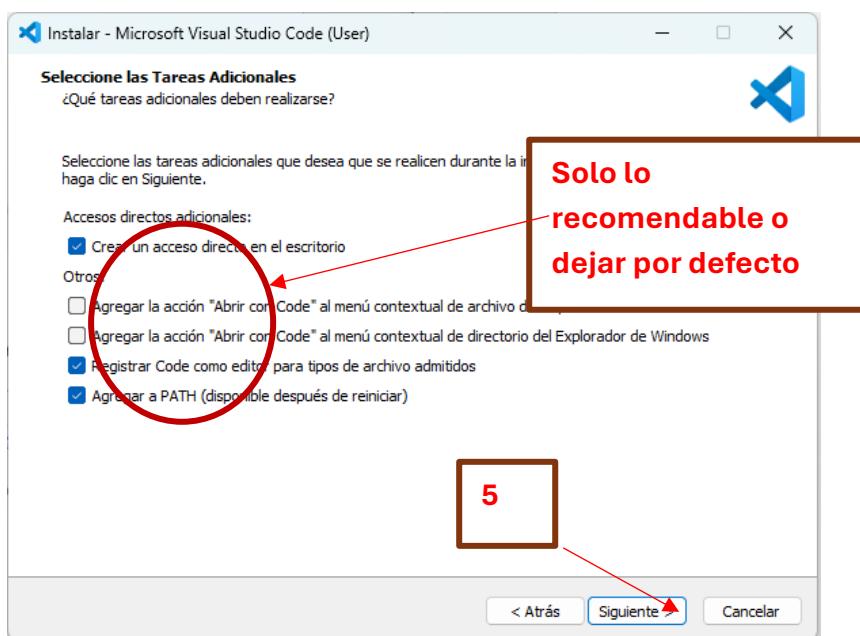
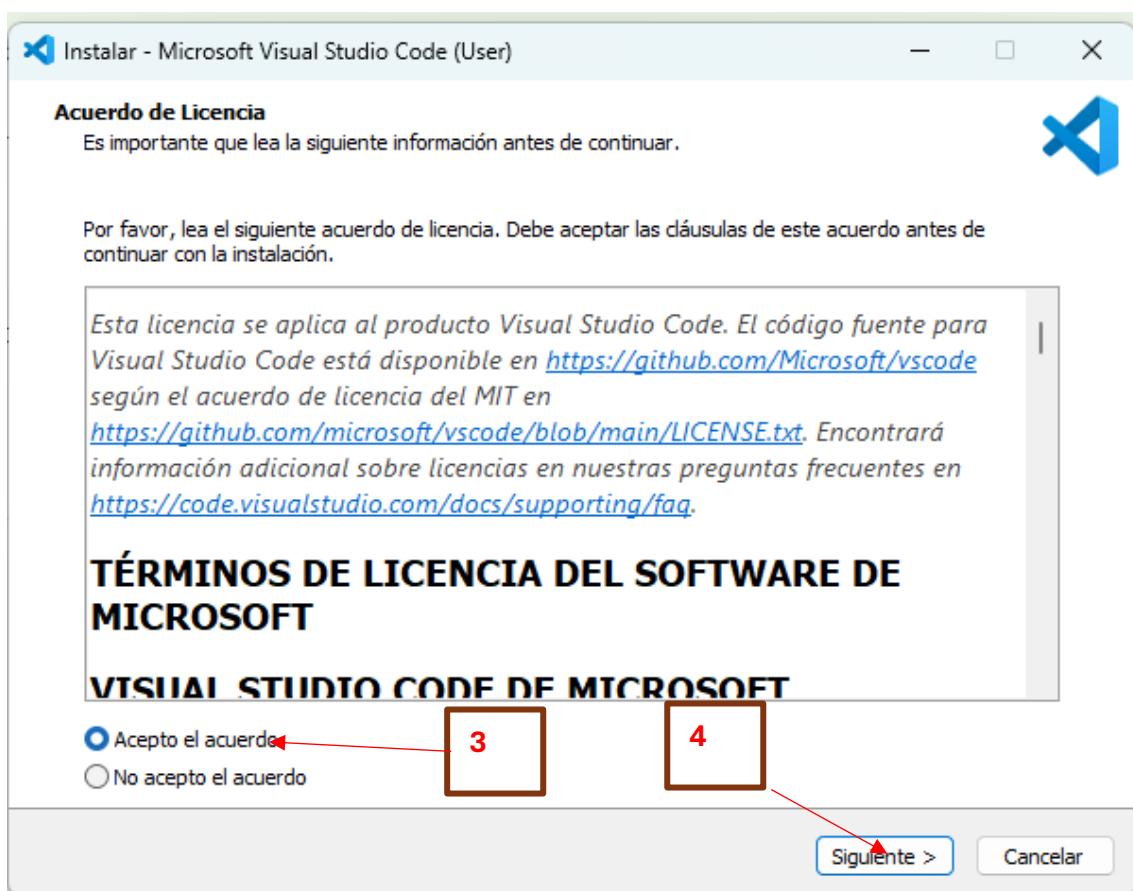
Visual Studio Code con GitHub Copilot 1 carga tu código con sugerencias impulsadas por IA, directamente en tu editor.

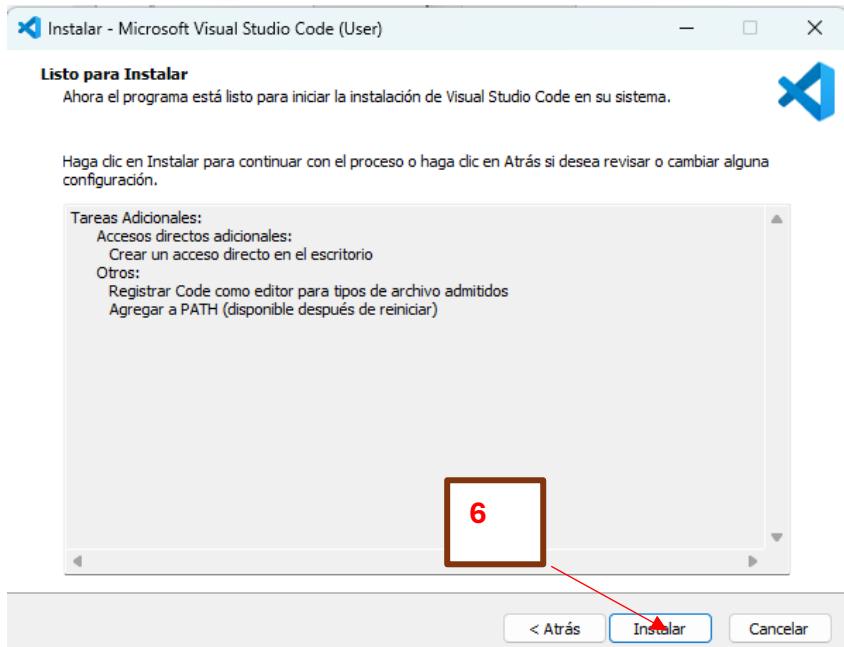
[Descargar para Windows](#) [Prueba GitHub Copilot](#)

[Web](#), [edición Insiders](#) u [otras plataformas](#)



- Acepta los términos y continua:

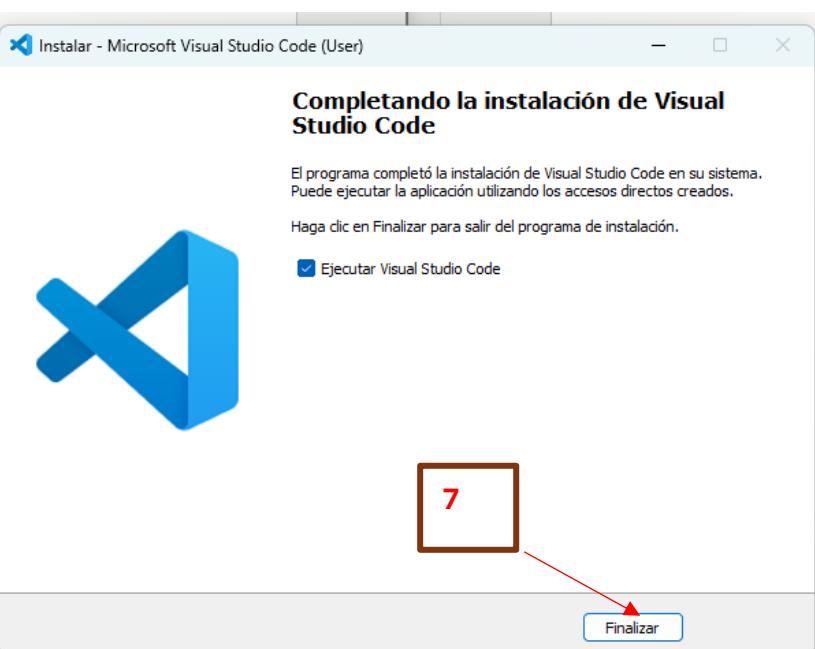




- Esperar que cargue el progreso de instalación y luego clic en finalizar:

Extrayendo archivos...

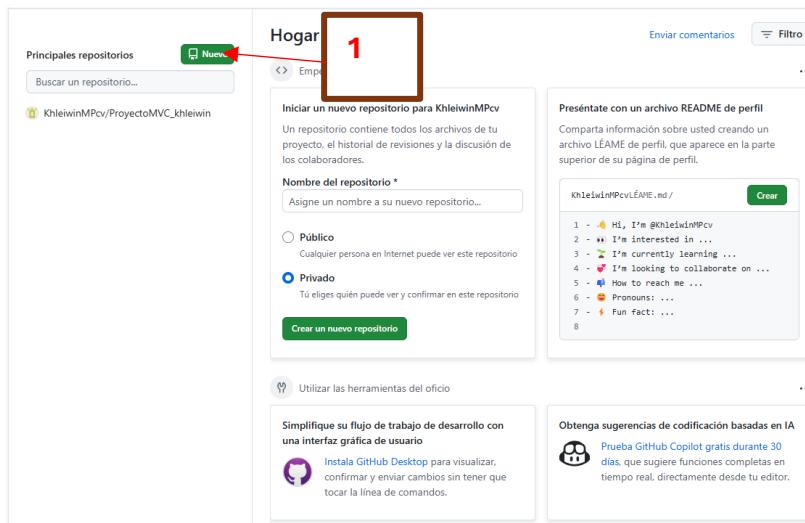
C:\Users\villa\AppData\Local\Programs\Microsoft VS Code\Code.exe



## Sección 2:Procedimiento de configuración de Jenkins para que realice la sincronización entre un proyecto en GitHub y el repositorio en el cliente (PC):

### 2.1. Creación del repositorio en GitHub:

- Para poder realizar la integración, necesitamos tener un proyecto en GitHub. Lo primero que debemos hacer es crear una cuenta en <https://github.com>. Puedes hacerlo con una cuenta Google que ya tengas. Una vez hecho esto, te dirigirá a la página principal, en donde debes dar clic en la opción ‘Nuevo’. Ubicado en la parte superior izquierda.



- Ahora, coloca un nombre al repositorio, selecciona que será privado (para mayor seguridad) y agrega un ‘README’. Para terminar, da clic en ‘crear repositorio’.

**Crear un nuevo repositorio**

Un repositorio contiene todos los archivos del proyecto, incluido el historial de revisiones. ¿Ya tienes un repositorio de proyectos en otro lugar? [Importar un repositorio](#).

Los campos obligatorios están marcados con un asterisco (\*).

Dueño *	Nombre del repositorio *
KhleiwInMPcv	MVC_Login

Los nombres de los grandes repositorios son cortos y fáciles de recordar. ¿Necesitas inspiración? ¿Qué tal Robot-Plata?

Descripción (opcional)

Público Cualquier persona en Internet puede ver este repositorio. Tú eliges quién puede comprometerse.

Privado Tú eliges quién puede ver y confirmar en este repositorio.

Inicialice este repositorio con:

Agregar un archivo README Aquí es donde puedes escribir una descripción larga para tu repositorio. [Más información sobre los archivos README](#)

Agregar .gitignore

Plantilla .gitignore: Ninguno

Elija los archivos de los que no se desea realizar un seguimiento de una lista de plantillas. [Obtén más información sobre cómo ignorar archivos](#)

Elija una licencia

Licencia: Ninguno

Una licencia le dice a otros lo que pueden y no pueden hacer con su código. [Más información sobre las licencias](#).

Este establecerá [principal](#) como la rama predeterminada. Cambie el nombre predeterminado en su configuración.

① Estás creando un repositorio privado en tu cuenta personal.

**5**

**4**

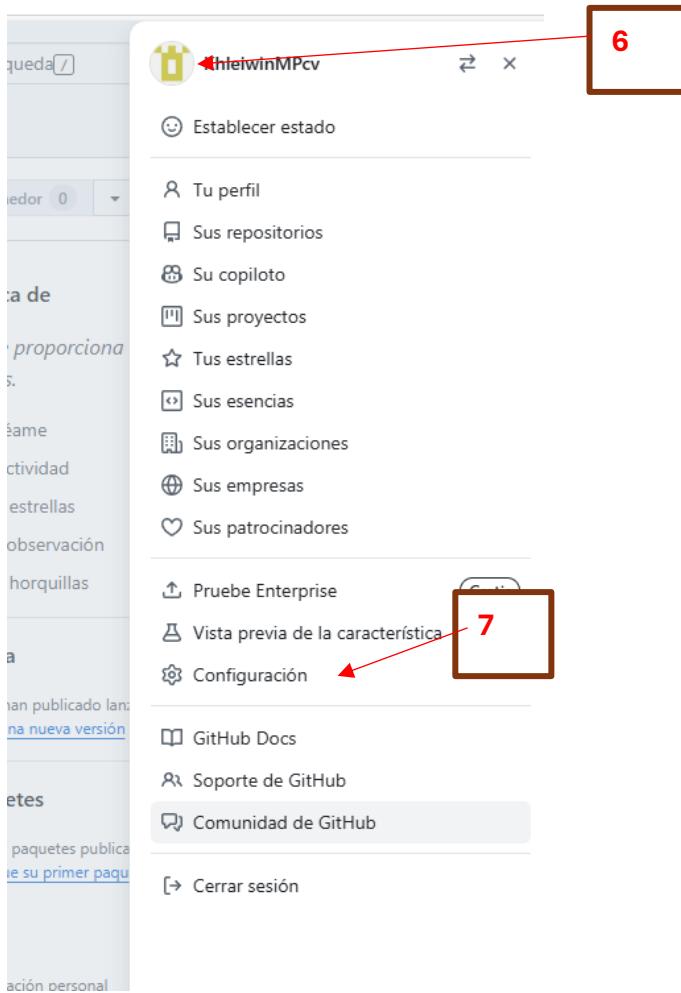
**3**

**2**

**opcional**

**Crear repositorio**

- Ahora, lo que tienes que hacer es crear un Token de Acceso Personal (PAT). Esto va a servir tanto para clonar nuestro repositorio en cualquier cliente como en nuestro servidor de Jenkins para la integración continua. Para esto accede a la configuración de desarrollador mediante las opciones principales “configuración” en tu perfil de usuario.



- Luego desplázate hasta encontrar la opción **Configuración del desarrollador** “clic”:

Moderación

Código, planeación y automatización

- Repositorios**
- Codespaces**
- Paquetes**
- Copiloto**
- Páginas**
- Respuestas guardadas**

Seguridad

- Seguridad del código**

Integraciones

- Aplicaciones**
- Reactorios programados**

Archivo

- Registro de seguridad**
- Registro de patrocinios**

<> **Configuración del desarrollador**

Pronombres

No especificar

URL

ORCID ID

ORCID proporciona un identificador persistente, un ORCID ID, que lo distingue de otros investigadores. Más información en [ORCID.org](#)

Conecta su ORCID ID

Cuentas sociales

Enlace al perfil social

Enlace al perfil social

Enlace al perfil social

Enlace al perfil social

Configuración

Número de horas

Última actualización

Mostrar la hora local actual

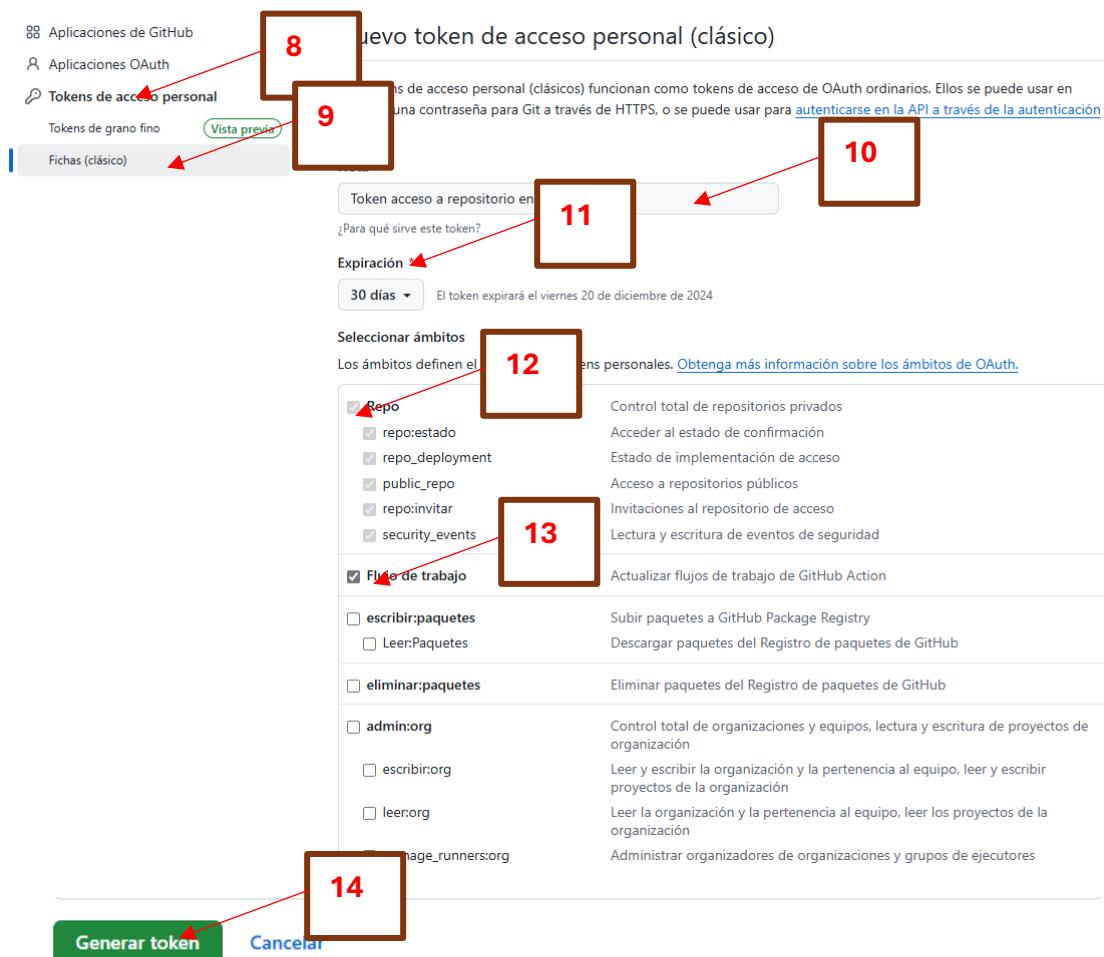
Los cambios de esta página son opcionales y se pueden eliminar en cualquier momento. Tenga en cuenta que si comparte este token con otro usuario, no estará dando permisos para que comparte este dato dondequiera que aparezca su perfil de usuario. Consulte nuestra [declaración de privacidad](#) para obtener más información sobre cómo utilizamos esta información.

Perfil de actualización

Crea un PAT clásico. Antes a esto nos pedirá crear un Token lo cual primero pedirá ingresar contraseña nuevamente de GitHub.

- Ingresamos un nombre que describa el uso de este Token e indicamos el tiempo de duración. Se recomienda no seleccionar ‘No expiración’ por temas de seguridad. En este caso seleccionaremos una expiración

de 30 días. Además configuramos los permisos y características de los ámbitos. Finalmente, clic en crear repositorio.



- Por temas de seguridad, el Token se mostrará una única vez. Copia y guarda el enlace y guarda en un lugar seguro:

## Tokens de acceso personal (clásico)

Generar nuevo token ▾

Tokens que has generado y que se pueden usar para acceder a la [API de GitHub](#).

⚠ Asegúrese de copiar su token de acceso personal ahora. ¡No podrá verlo más tarde!

15 

✓ ghp\_n...naOKIcSgShUeoZpkz1NYAej3MZend2GvduB  

Los tokens de acceso personal (clásicos) funcionan como tokens de acceso de OAuth ordinarios. Ellos se puede usar en lugar de una contraseña para Git a través de HTTPS, o se puede usar para [autenticarse en la API a través de la autenticación básica](#).

- Una vez hecho esto, debemos subir nuestro proyecto al repositorio que hemos creado. Accedemos a este mediante la página principal.

### Principales repositorios

 Nuevo

Buscar un repositorio...

16 

 KhleiwinMPcv/ProyectoMVC\_khleiwin

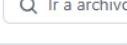
 KhleiwinMPcv/MVC\_Loguin

- En la parte superior aparecerá nuestro usuario y el nombre del repositorio. Copiamos el usuario y damos clic en ‘Código’. En la pestaña de ‘Https’, copiamos el enlace del repositorio.

17 

MVC\_Loguin

18 

principal 1 Rama  Código

KhleiwinMPcv Confirmación Local Codespaces

19  20 

https://github.com/KhleiwinMPcv/MVC\_Loguin.git Clonar usando la URL web.

Abrir con GitHub Desktop Descargar ZIP

- Ingresamos al **git bash** y configuramos las credenciales para poder subir nuestro proyecto al repositorio. Ingresamos la siguientes lineas, reemplazando por los datos que ya hemos guardado anteriormente (OJO: el mail es el correo con el que creamos nuestra cuenta de GitHub):

⇒ **git config --global user.name "tu-usuario"**  
 ⇒ **git config --global user.password "tu-token"**  
 ⇒ **git config --global user.mail "tu-correo"**

```
villa@DESKTOP-DRJM31E MINGW64 ~
$ git config --global user.name "khleiwinMPcv"
villa@DESKTOP-DRJM31E MINGW64 ~
$ khleiwinMPcv
bash: khleiwinMPcv: command not found
villa@DESKTOP-DRJM31E MINGW64 ~
$ git config --global user.khleiwinMPcv
villa@DESKTOP-DRJM31E MINGW64 ~
$ git config --global user.password "ghp_ra0XKICShUeoZpkz1NYAej3MZend2C...
villa@DESKTOP-DRJM31E MINGW64 ~
$ git config --global user.mail "villaloboskhleiwin@gmail.com"
villa@DESKTOP-DRJM31E MINGW64 ~
$
```

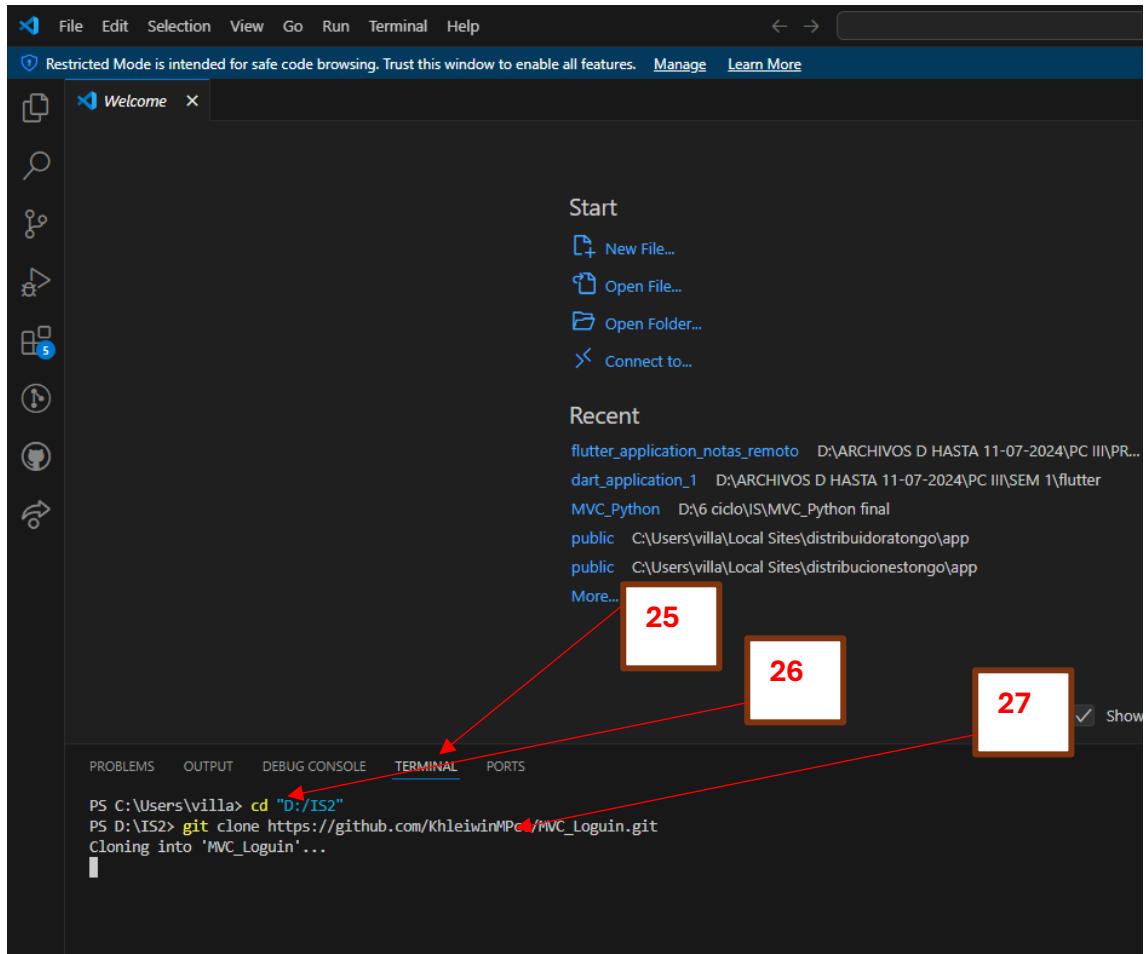
**Importante:** Los comandos utilizados con ‘GitHub’ son los mismos para cualquier sistema operativo.

Ahora, elegimos una carpeta en donde deseemos iniciar el proceso de subida de nuestro proyecto. Para esto, nos apoyaremos de **visual studio Code**. Abrimos y nos dirigimos a la terminal que nos proporciona la propia aplicación e ingresamos el siguiente comando:

⇒ **cd “dirección/de/la/carpeta/donde/clonar/repositorio”**

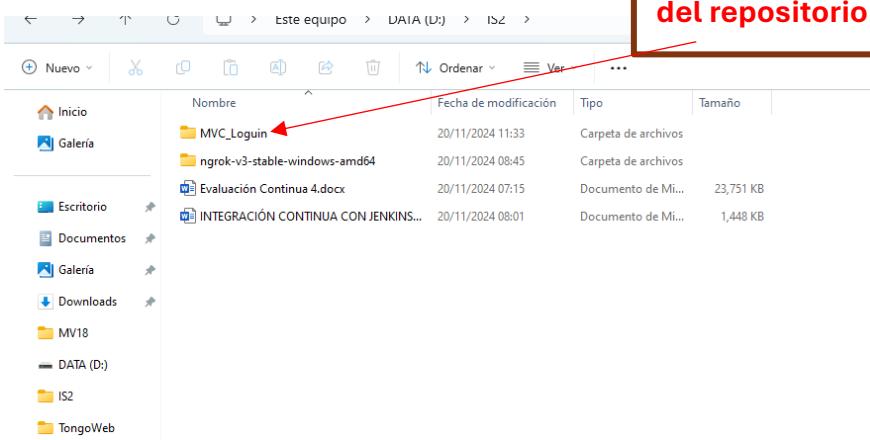
Este comando nos dirigirá a la carpeta en donde queramos clonar el repositorio. Ahora, ingresamos este otro comando, utilizando el enlace del repositorio que copiamos en el **paso 20**:

⇒ **git clone (enlace del repositorio del paso 20)**

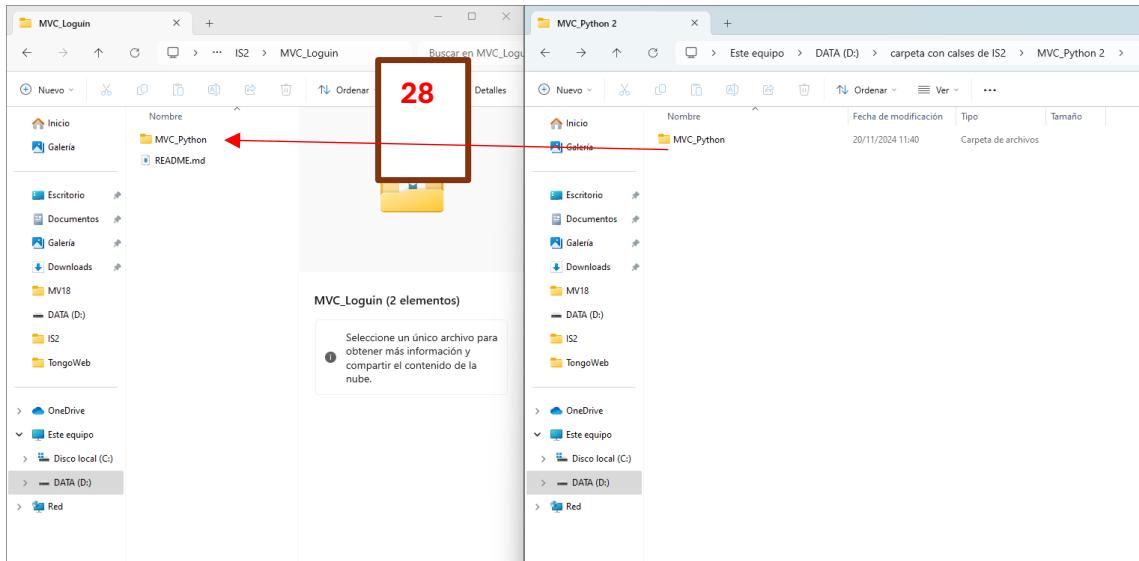


- Se habrá creado una carpeta con el nombre del repositorio en la dirección que hemos ingresado. Ahora arrastramos a esta misma carpeta el proyecto que queremos:

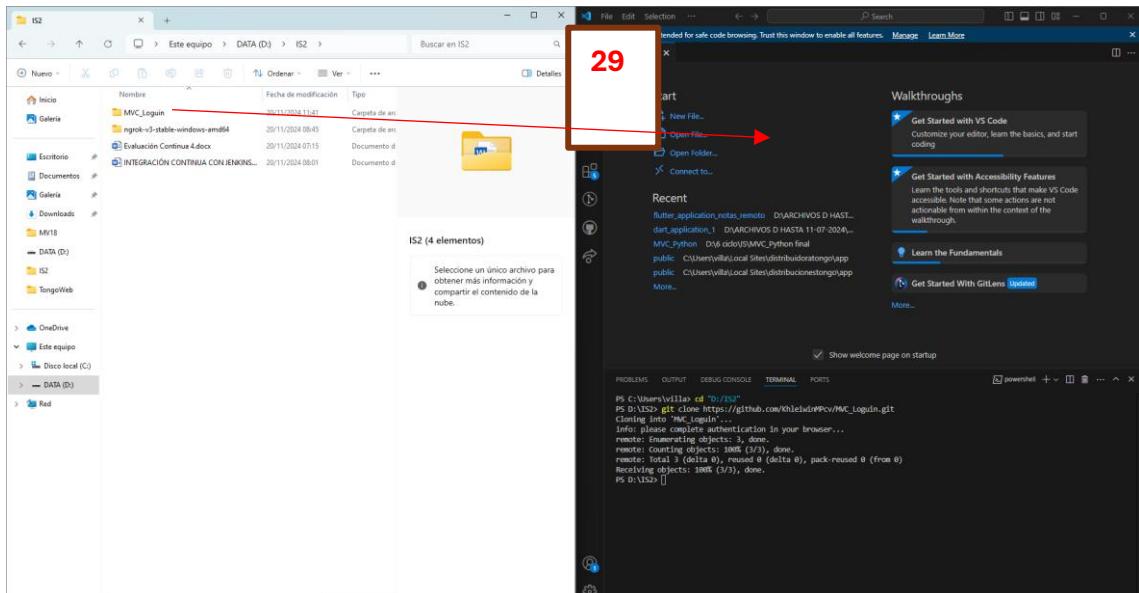
**Carpeta con nombre del repositorio**



- Ahora, copiamos y pegamos a esta misma carpeta la carpeta del proyecto que queremos subir.



- Después, arrastramos o abrimos con la opción **open folder** la carpeta a **Visual Studio Code** ya con el proyecto dentro.

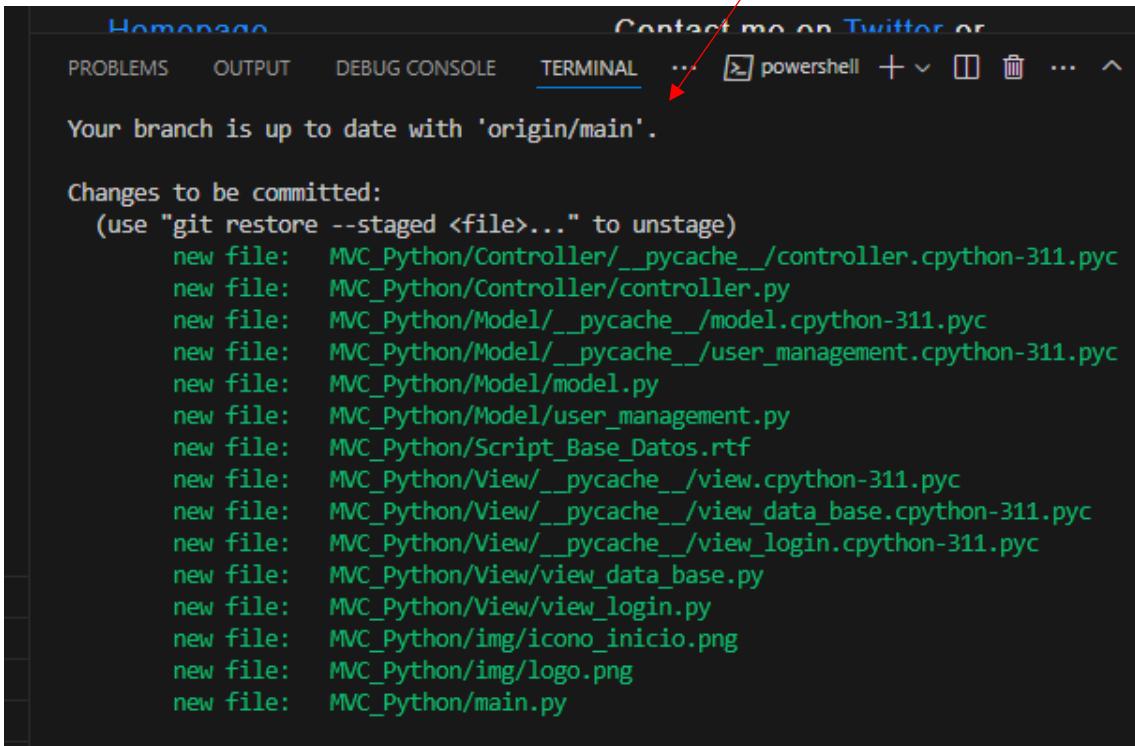


- En Visual Studio Code, accedemos nuevamente al terminal y ejecutamos el siguiente comando:  
⇒ **git add carpeta\_del\_proyecto/**

```
PS D:\IS2\MVC_Loguin> git add MVC_Python/
warning: in the working copy of 'MVC_Python\Controller\controller.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'MVC_Python\Model\model.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'MVC_Python\Model\user_management.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'MVC_Python\Script\Run_Program.py', LF will be replaced by CRLF the next time Git touches it
```

- Verificamos que no se haya generado ningún error con el siguiente comando en el terminal de Visual Studio Code: **git status**

30



```
Your branch is up to date with 'origin/main'.

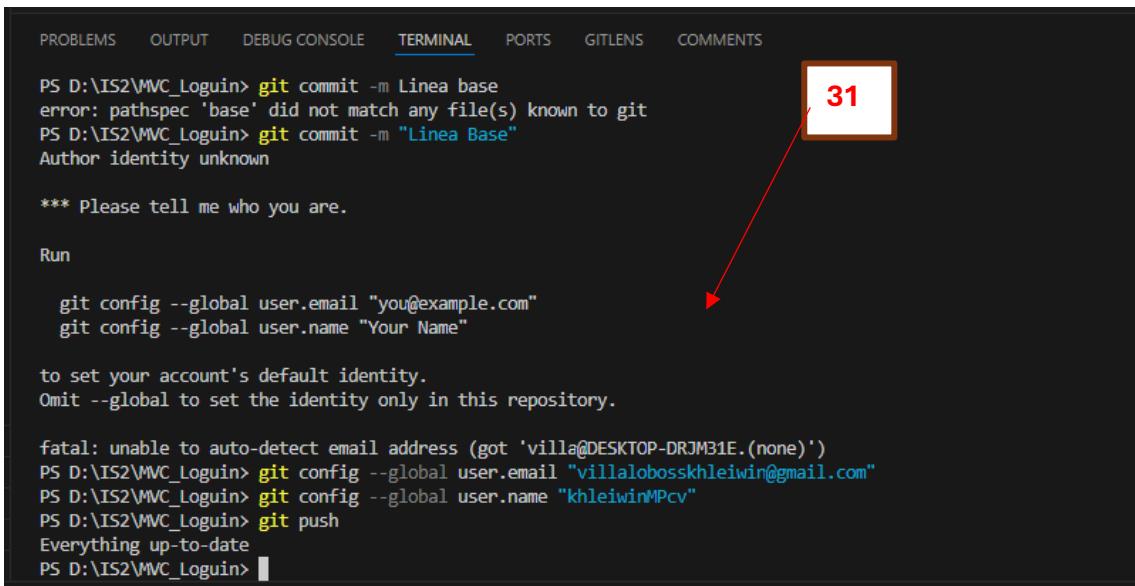
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  MVC_Python/Controller/__pycache__/controller.cpython-311.pyc
    new file:  MVC_Python/Controller/controller.py
    new file:  MVC_Python/Model/__pycache__/model.cpython-311.pyc
    new file:  MVC_Python/Model/__pycache__/user_management.cpython-311.pyc
    new file:  MVC_Python/Model/model.py
    new file:  MVC_Python/Model/user_management.py
    new file:  MVC_Python/Script_Base_Datos.rtf
    new file:  MVC_Python/View/__pycache__/view.cpython-311.pyc
    new file:  MVC_Python/View/__pycache__/view_data_base.cpython-311.pyc
    new file:  MVC_Python/View/__pycache__/view_login.cpython-311.pyc
    new file:  MVC_Python/View/view_data_base.py
    new file:  MVC_Python/View/view_login.py
    new file:  MVC_Python/img/icono_inicio.png
    new file:  MVC_Python/img/logo.png
    new file:  MVC_Python/main.py
```

- Por último, debemos ingresar los siguientes dos comandos para concretar con la subida del proyecto al repositorio. En **Línea Base** reemplazar si desea otro nombre.

⇒ **git commit -m “Línea base”**

⇒ **git push**

31



```
PS D:\IS2\MVC_Login> git commit -m Línea base
error: pathspec 'base' did not match any file(s) known to git
PS D:\IS2\MVC_Login> git commit -m "Línea Base"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'villa@DESKTOP-DRJM31E.(none)')
PS D:\IS2\MVC_Login> git config --global user.email "villaloboskhleiwin@gmail.com"
PS D:\IS2\MVC_Login> git config --global user.name "khleiwinMPcv"
PS D:\IS2\MVC_Login> git push
Everything up-to-date
PS D:\IS2\MVC_Login>
```

- Se puede comprobar que el proyecto se subió al repositorio accediendo a él con el navegador de su preferencia.

MVC\_Loguin (Privado)

Dejar de ver 1

principal 1 Rama 0 Etiquetas

Ir a archivo Agregar archivo Código

KhleiwinMPcv	Línea base	AA4463B4 · 1 minute ago
MVC_Python	Agregar archivos a través de la carga	10 minutes ago
Línea Base	Crear Línea Base	4 minutes ago
README.md	Confirmación inicial	2 hours ago

LÉAME

MVC\_Loguin

## 2.2. Creación del WebHook:

Si deseamos hacer que Jenkins integre continuamente cualquier cambio en nuestro repositorio, es necesario crear un WebHook para que GitHub notifique a Jenkins de manera constante, haciendo que el Job (el cuál crearemos más adelante) se realice de manera automática.

- Para esto, debemos acceder a la configuración de nuestro repositorio, luego dirigirse a la sección de “Webhook” para presionar el botón agregar **webHook**

KhleiwinMPcv / MVC\_Login ð

Configuración

General Webhooks

Acceso

Colaboradores

Código y automatización

Ramas

Etiquetas

Reglas

Acciones

**Webhooks**

CodeSpaces

Páginas

Seguridad

Seguridad del código

Implementación de claves

Secretos y variables

Integraciones

Aplicaciones de GitHub

Notificaciones por correo electrónico

Agregar webhook

- Pegaremos el URL remota que creamos en **el paso 8 de la Descargar NGROK** en el campo ‘Payload URL’ agregando el texto ‘/github-webhook/’. Terminamos la creación del Webhook haciendo clic en ‘**Agregar WebHook**’.

Webhooks / Agregar webhook

Enviremos una solicitud a la URL que aparece a continuación con los detalles de los eventos suscritos. También puede especificar el formato de datos que desea recibir (JSON, , etc). Puede encontrar más información en [nuestra documentación para desarrolladores](#). POSTx-www-form-urlencoded

URL de carga útil \*

https://rapid-vulture-forcibly.ngrok-free.app → http://localhost:8080

Tipo de contenido \*

application/x-www-form-urlencoded

Secreto

Verificación SSL

De forma predeterminada, verificamos los certificados SSL al entregar cargas útiles.

Habilitar la verificación  SSL Desactivar (no recomendado)

¿Qué eventos te gustaría que desencadenen este webhook?

Solo el evento de empuje.

Mándame todo.

Permítanme seleccionar eventos individuales.

Activo

Entregaremos los detalles del evento cuando \_\_\_\_\_ ganchos.

Agregar webhook

- Esperar unos segundos para que se agregue y finalmente veremos esto:

The screenshot shows the Jenkins Webhooks configuration page. At the top right is a button labeled "Aregar webhook". Below it is a section with the heading "Webhooks" and a sub-section "Guía de webhooks". A red box labeled "6" highlights a URL entry field containing "https://rapid-vulture-forci... (empujar)". To the left of the URL is a green checkmark icon. To the right are "Editar" and "Borrar" buttons. Below the URL field is the message "La última entrega fue exitosa.".

Lo cual significa que se agregó y configuro correctamente.

### 2.3. Creación del Job (Jenkins):

Ahora, pasaremos a crear el Job. Un Job en Jenkins es una unidad de trabajo que define un conjunto de tareas o pasos específicos que Jenkins debe ejecutar para realizar un trabajo particular. En este caso, será un Job que se ejecutará cada vez que haya un cambio en nuestro repositorio en GitHub.

- Para crear nuestro Job nos dirigimos a la pestaña principal de nuestro Jenkins y seleccionamos la opción “Nueva tarea”. Una vez dentro, nos pedirá asignarle un nombre y debemos seleccionar la opción de ‘Crear un proyecto de estilo libre’. Apretamos ‘Ok’.

Panel de Control >

**1**

+ Nueva Tarea 

Historial de trabajos

Administrar Jenkins

Mis vistas

**Bienvenido a Jenkins!**

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Trabajos en la cola 

No hay trabajos en la cola

Estado del ejecutor de construcciones 0/2 

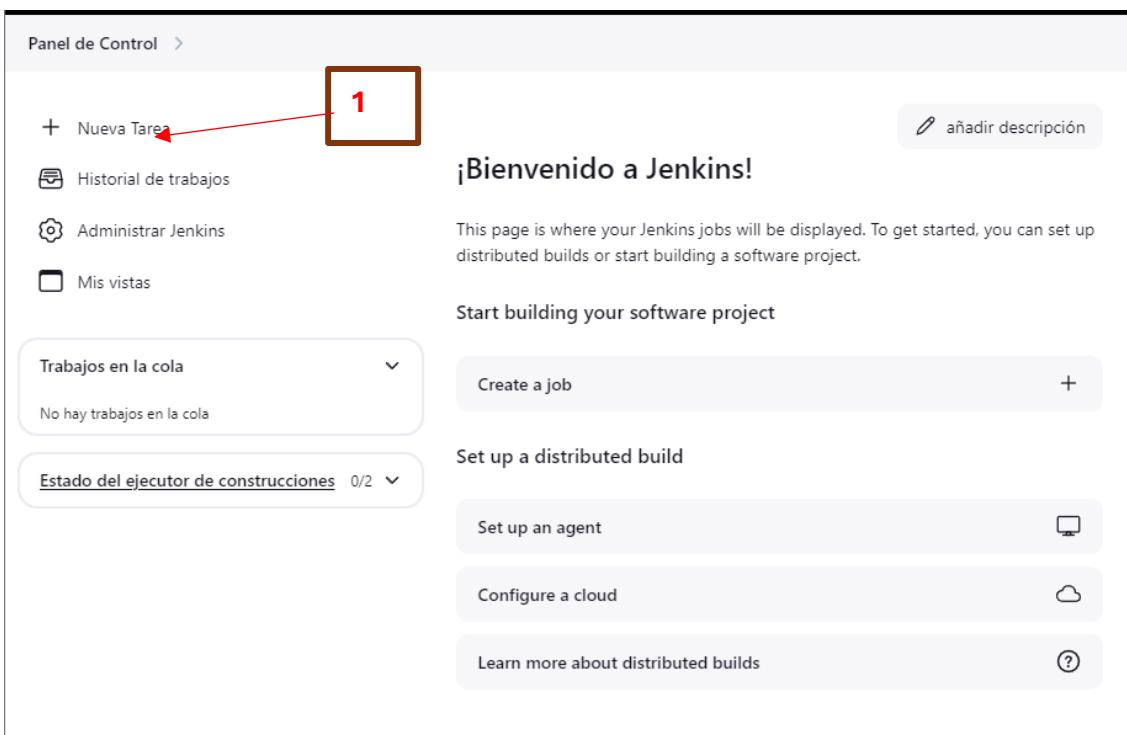
Create a job 

Set up a distributed build

Set up an agent 

Configure a cloud 

Learn more about distributed builds 



## Nuevo Tarea

Enter an item name

job1

2

Select an item type



Crear un proyecto de estilo libre

Classic, general-purpose job type that checks out from up to one SCM, executes steps like archiving artifacts and sending email notifications.

3



Pipeline

Gestiona actividades de larga duración que pueden abarcar varios agentes pipelines (conocidas anteriormente como workflows) y/o para la organización articulante fácilmente con tareas de tipo freestyle.



Crear un proyecto multi-configuración

Adecuado para proyectos que requieran un gran número de configuraciones en entornos, ejecutar sobre plataformas concretas, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. A folder creates a separate namespace, so you can have multiple things of the same name in different folders.



Multibranch Pipeline

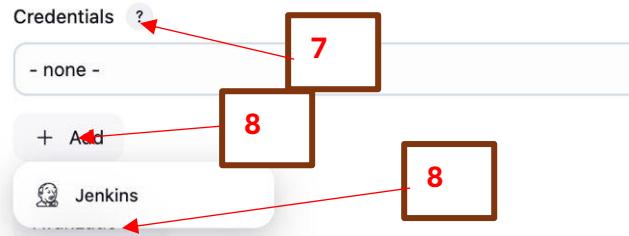
Creates a set of Pipeline projects according to detected branches in one SCM.

4

- Una vez en la configuración del **Job en Jenkins**, dirígete al apartado de ‘Configurar el origen del código fuente’ y seleccionamos la opción de ‘Git’. Nos pedirá pegar el URL del repositorio.



- Después, debemos configurar las credenciales. En la parte siguiente encontraremos el campo “Credentials” presionamos en la pregunta. Luego, presionamos ‘Add’ y ‘Jenkins’.



- Dejar en la configuración predeterminada y en ‘Username’ ingresaremos nuestro usuario de GitHub. En ‘Password’, el Token de Acceso Personal. En ‘ID’ colocaremos ‘GitHub’ y en ‘Description’, ‘Credenciales de GitHub’. Finalmente, terminaremos presionando ‘Add’.

Jenkins Credentials Provider: Jenkins

Scope ? Global (Jenkins, nodes, items, all child items, etc)

Username ? KhleiwinMPcv 9

Treat username as secret ?

Password ? ..... 10

ID ? GitHub 11

Description ? credenciales de GitHub 12

Cancel Add 13

- Lugo ya seleccionamos las credenciales que se registro:

Credentials ?

KhleiwinMPcv/\*\*\*\*\* (credenciales de GitHub) 14

+ Add

- En 'Branches to build', nos aparecerá por defecto '`*/master`', lo cambiaremos a '`*/main`'.

Branches to build ?

Branch Specifier (blank for 'any') ?

\*/main

Add Branch

- ¿En la sección de 'Triggers?', debemos activar solo la opción '**GitHub hook trigger for GITScm polling**'. Esta es la que nos permitirá hacer la integración continua permanentemente con cambios.

**Triggers**

Set up automated actions that start your build based on specific changes or scheduled times.

- Lanzar ejecuciones remotas (ejem: desde 'scripts')
- Construir tras otros proyectos ?
- Consultar repositorio (SCM) ?
- Ejecutar periódicamente ?
- GitHub hook trigger for GITScm polling ?

- En la sección de 'Build Steps', seleccionaremos la opción 'Ejecutar un comando Windows', ya que nuestro servidor de Jenkins está sobre este SO. Por ejemplo, si fuera otro SO, elegir la opción de 'Ejecutar linea de comandos (shell)', este para Linux.

## Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Añadir un nuevo paso ^

Filter

- Ejecutar Ant
- Ejecutar linea de comandos (shell)
- Ejecutar tareas 'maven' de nivel superior
- Ejecutar un comando de Windows** 17
- Invoke Gradle script
- Run with timeout
- Set build status to "pending" on GitHub commit

Ya que solamente nos interesa comprobar la integración continua entre nuestro repositorio y el servidor Jenkins, colocaremos un comando que nos confirme la realización del Job: **echo 'Integración con éxito'**. Después de esto, presionamos 'Save'.

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Ejecutar un comando de Windows 18

Comando

Visualiza la lista de variables de entorno disponibles

```
echo 'Integración con éxito'
```

Avanzado ▾

Añadir un nuevo paso ▾

### Acciones para ejecutar después.

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Añadir una acción ▾ 19

Save Apply

- Despues de eso si se realizó correctamente debe mostrar esto:

 job1



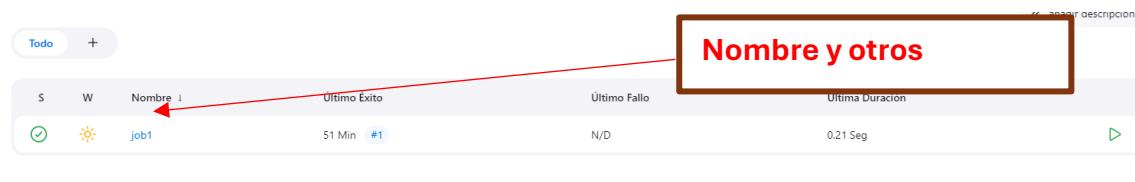
añadir descripción

## Enlaces permanentes

- "Última ejecución (#1) hace 46 Min"
- "Última ejecución estable (#1) hace 46 Min"
- "Última ejecución correcta (#1) hace 46 Min"
- "Last completed build (#1) hace 46 Min"

20

- En este punto del proceso, sería importante incluir los comandos y pasos necesarios para realizar tareas como la comprobación de errores y el despliegue. Sin embargo, el propósito de este manual es únicamente mostrar cómo se realiza la integración continua. El trabajo que hemos creado podrá verse en la pestaña principal de Jenkins. Esto se muestra en el panel de control de Jenkins



The screenshot shows the Jenkins dashboard with the 'Nombre y otros' card highlighted by a red box. The card displays the following information:

S	W	Nombre	Último Éxito	Último Fallo	Última Duración
		job1	51 Min #1	N/D	0.21 Seg

Icons at the bottom left: S, M, L. Top right: añadir descripción.

Nombre y otros

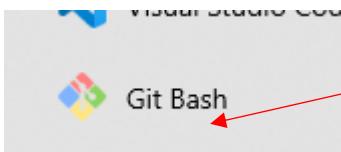
### Sección 3: Ejecución de la sincronización:

Para esto, utilizaremos dos computadoras, una que sirva como cliente y la otra como servidor, que es de hecho en donde hemos instalada el Jenkins.

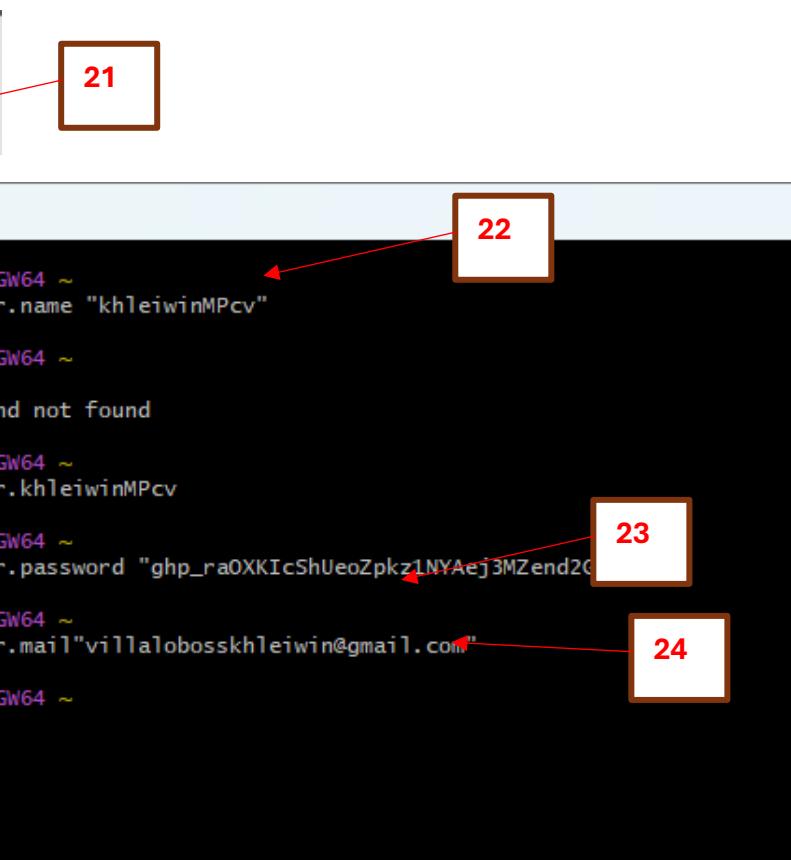
#### 3.1. Para ello debemos repetir esto:

Ingresamos al **git bash** y configuramos las credenciales para poder subir nuestro proyecto al repositorio. Ingresamos la siguientes líneas, reemplazando por los datos que ya hemos guardado anteriormente (OJO: el mail es el correo con el que creamos nuestra cuenta de GitHub):

```
⇒ git config --global user.name "tu-usuario"  
⇒ git config --global user.password "tu-token"  
⇒ git config --global user.mail "tu-correo"
```



The screenshot shows the Visual Studio Code interface. A red arrow points from a red box labeled '21' to the 'Git Bash' icon in the sidebar.



The terminal window has a red box labeled '22' pointing to the command line. It displays the following text:

```
villa@DESKTOP-DRJM31E MINGW64 ~  
$ git config --global user.name "khleiwinMPcv"  
  
villa@DESKTOP-DRJM31E MINGW64 ~  
$ khleiwinMPcv  
bash: khleiwinMPcv: command not found  
  
villa@DESKTOP-DRJM31E MINGW64 ~  
$ git config --global user.khleiwinMPcv  
  
villa@DESKTOP-DRJM31E MINGW64 ~  
$ git config --global user.password "ghp_ra0XKICShUeoZpkz1NYAej3MZend2C  
  
villa@DESKTOP-DRJM31E MINGW64 ~  
$ git config --global user.mail "villaloboskhleiwin@gmail.com"  
  
villa@DESKTOP-DRJM31E MINGW64 ~  
$
```

Red arrows point from boxes labeled '23' and '24' to the password and email configuration lines respectively.

**Importante:** Los comandos utilizados con ‘GitHub’ son los mismos para cualquier sistema operativo.

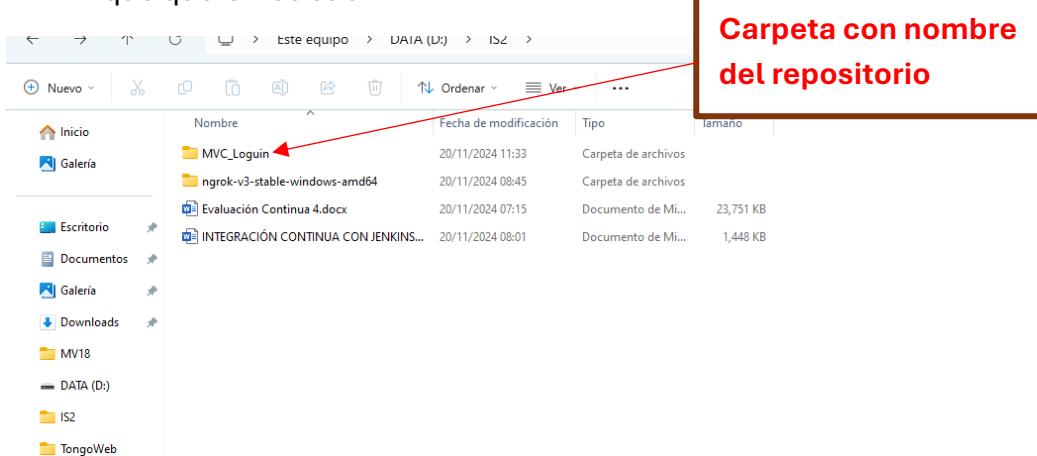
- Ahora, elegimos una carpeta en donde deseemos iniciar el proceso de subida de nuestro proyecto. Para esto, nos apoyaremos de **visual studio Code**. Abrimos y nos dirigimos a la terminal que nos proporciona la propia aplicación e ingresamos el siguiente comando:  
⇒ **cd “dirección/de/la/carpeta/donde/clonar/repositorio”**

- Este comando nos dirigirá a la carpeta en donde queramos clonar el repositorio. Ahora, ingresamos este otro comando, utilizando el enlace del repositorio que copiamos en el **paso 20**:

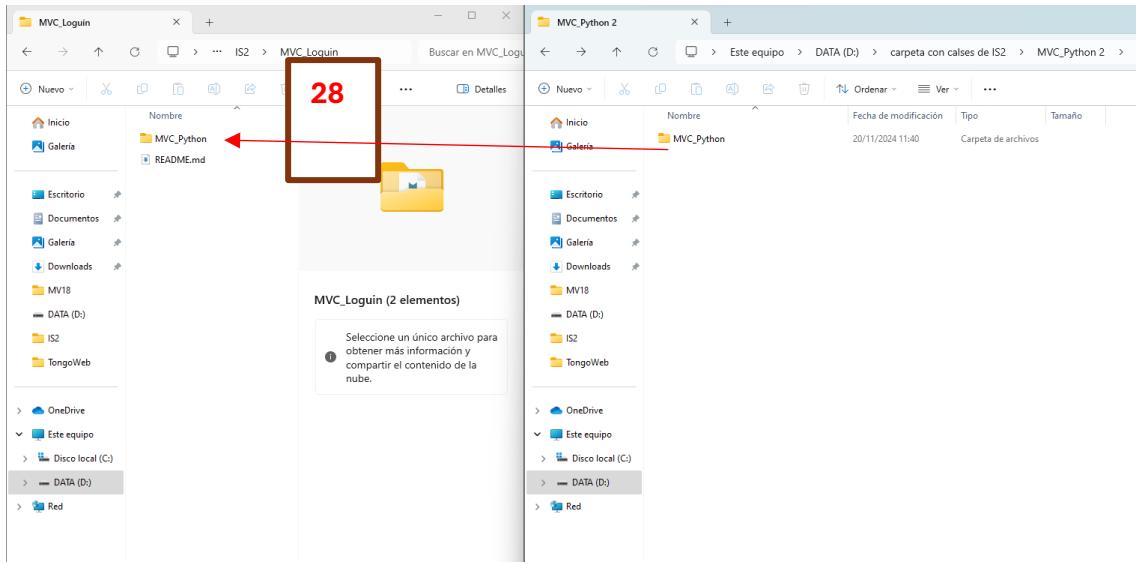
⇒ **git clone (enlace del repositorio del paso 20)**

```
PS C:\Users\villa> cd "D:\IS2"
PS D:\IS2> git clone https://github.com/KhleiwinMPc/MVC_Login.git
Cloning into 'MVC_Login'...
```

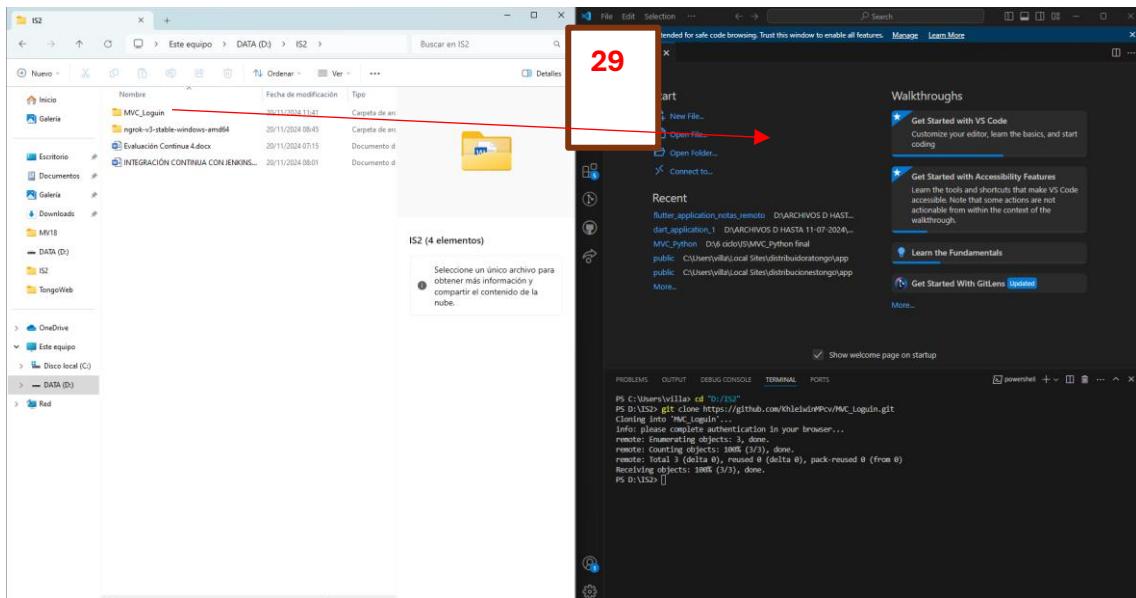
- Se habrá creado una carpeta con el nombre del repositorio en la dirección que hemos ingresado. Ahora, arrastramos a esta misma carpeta el proyecto que queremos subir.



- Ahora, copiamos y pegamos a esta misma carpeta la carpeta del proyecto que queremos subir.



- Despues, arrastramos o abrimos con la opción **open folder** la carpeta a **Visual Studio Code** ya con el proyecto dentro.



- En Visual Studio Code, accedemos nuevamente al terminal y ejecutamos el siguiente comando:  
⇒ **git add carpeta\_del\_proyecto/**

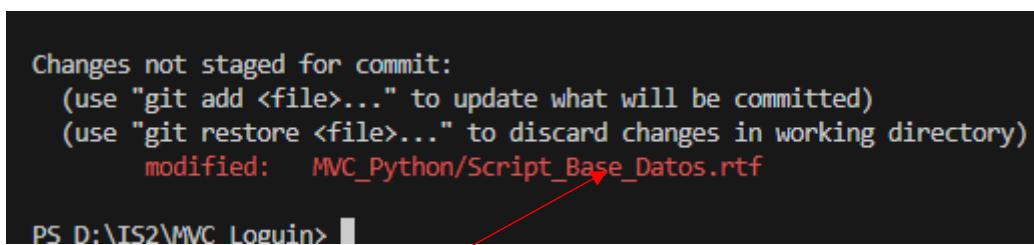


```
PS D:\IS2\MVC_Login> git add MVC_Python/
warning: in the working copy of 'MVC_Python\Controller/controller.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'MVC_Python\Model/model.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'MVC_Python\Model/user_management.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'MVC_Python\Script_Base_Datos.rtf', LF will be replaced by CRLF the next time Git touches it
```

- Procederemos a realizar algún cambio en el código para posteriormente realizar su respectivo **git push**:
- Luego de eso nos muestra el ultimo cambio en:



- Modifico un archivo o el código de alguno. Al ejecutar el **git Status**



```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   MVC_Python/Script_Base_Datos.rtf

PS D:\IS2\MVC_Login>
```

Y en este archivo se realizó el cambio

### 3.2. Gestión cambios

Actualizamos dependencias: (ESTO ES SOLO PARA EL PROYECTO DE FLUTTER EN VScODE)

```
PS D:\ARCHIVOS D HASTA 11-07-2024\PC III\PROYECTO FINAL PC III\notas Remoto\notas Remoto\flutter_application_notas_remoto> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.24.5, on Microsoft Windows [Versión 10.0.26100.2454], locale es-PE)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.11.6)
[✓] Android Studio (version 2023.3)
[✓] VS Code (version 1.95.3)
[✓] Connected device (4 available)
[✓] Network resources
```

Inicializamos el proyecto par GITUB:

```
PS D:\ARCHIVOS D HASTA 11-07-2024\PC III\PROYECTO FINAL PC III\notas Remoto\notas Remoto\flutter_application_notas_remoto> git init
Initialized empty Git repository in D:/ARCHIVOS D HASTA 11-07-2024/PC III/PROYECTO FINAL PC III/notas Remoto/notas Remoto/flutter_application_notas_remoto/.git/
PS D:\ARCHIVOS D HASTA 11-07-2024\PC III\PROYECTO FINAL PC III\notas Remoto\notas Remoto\flutter_application_notas_remoto> git add .
warning: in the working copy of '.metadata', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'devtools/options.yaml', LF will be replaced by CRLF the next time Git touches it
warning: adding embedded git repository: flutter
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint: git submodule add <url> flutter
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint: git rm --cached flutter
hint:
hint: See "git help submodule" for more information.
hint: Disable this message with "git config advice.addEmbeddedRepo false"
warning: in the working copy of 'linux/flutter/generated_plugin_registrant.cc', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'linux/flutter/generated_plugin_registrant.h', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'linux/flutter/generated_plugin_registrant.mak', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'macos/flutter/generatedPluginRegistrant.swift', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'pubspec.lock', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'windows/flutter/generated_plugin_registrant.cc', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'windows/flutter/generated_plugin_registrant.h', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'windows/flutter/generated_plugins.cmake', LF will be replaced by CRLF the next time Git touches it
```

Levantamos el proyecto:

```
PS D:\ARCHIVOS D HASTA 11-07-2024\PC III\PROYECTO FINAL PC III\notas Remoto\notas Remoto\flutter_application_notas_remoto> git commit -m "Primer commit"
[master (root-commit) 6a558d8] Primer commit
170 files changed, 8062 insertions(+)
create mode 100644 .gitignore
create mode 100644 .metadata
create mode 100644 .vs/ChakraWorkspaceSettings.json
create mode 100644 .vs/ProjectSettings.json
create mode 100644 .vs/VSWorkspaceState.json
create mode 100644 .vs/flutter_application_notas_remoto/fileContentIndex/7b64bf30-5e24-4f22-916e-bfe70e936188.vsidx
create mode 100644 .vs/flutter_application_notas_remoto/v17/.wsuo
create mode 100644 .vs/flutter_application_notas_remoto/v17/Browse.VC.db
create mode 100644 .vs/flutter_application_notas_remoto/v17/DocumentLayout.json
create mode 100644 .vs/flutter_application_notas_remoto/v17/workspaceFileList.bin
create mode 100644 .vs/slnx.sqlite
create mode 100644 .vscode/launch.json
create mode 100644 .vscode/settings.json
create mode 100644 README.md
create mode 100644 analysis_options.yaml
create mode 100644 android/.gitignore
create mode 100644 android/app/build.gradle
create mode 100644 android/app/src/debug/AndroidManifest.xml
create mode 100644 android/app/src/main/AndroidManifest.xml
create mode 100644 android/app/src/main/kotlin/com/example/flutter_application_notas_remoto/MainActivity.kt
create mode 100644 android/app/src/main/res/drawable-v21/launcher_background.xml
create mode 100644 android/app/src/main/res/drawable/launcher_background.xml
create mode 100644 android/app/src/main/res/mipmap-hdpi/ic_launcher.png
create mode 100644 android/app/src/main/res/mipmap-mdpi/ic_launcher.png
create mode 100644 android/app/src/main/res/mipmap-xhdpi/ic_launcher.png
create mode 100644 android/app/src/main/res/mipmap-xxhdpi/ic_launcher.png
create mode 100644 android/app/src/main/res/mipmap-xxxhdpi/ic_launcher.png
create mode 100644 android/app/src/main/res/values-night/styles.xml
create mode 100644 android/app/src/main/res/values/styles.xml
create mode 100644 android/app/src/profile/AndroidManifest.xml
```

```
PS D:\ARCHIVOS D HASTA 11-07-2024\PC III\PROYECTO FINAL PC III\notas Remoto\notas Remoto\flutter_application_notas_remoto> git remote add origin https://github.com/KhleiwinMPcv/ProyectoMVC_khleiwin.git
PS D:\ARCHIVOS D HASTA 11-07-2024\PC III\PROYECTO FINAL PC III\notas Remoto\notas Remoto\flutter_application_notas_remoto> git push -u origin master
Enumerating objects: 225, done.
Counting objects: 100% (225/225), done.
Delta compression using up to 16 threads.
Compressing objects: 100% (194/194), done.
Writing objects: 100% (225/225), 1.92 MiB | 1.08 MiB/s, done.
Total 225 (delta 30), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (30/30), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/KhleiwinMPcv/ProyectoMVC_khleiwin/pull/new/master
remote:
To https://github.com/KhleiwinMPcv/ProyectoMVC_khleiwin.git
 * [new branch]      master -> master
```

Verificamos el nuevo estado:

```
PS D:\ARCHIVOS D HASTA 11-07-2024\PC III\PROYECTO FINAL PC III\notas Remoto\notas Remoto\flutter_application_notas_remoto> git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      (commit or discard the untracked or modified content in submodules)
        modified:   flutter (modified content, untracked content)
```

Clonamos el repositorio en una carpeta de directorio local:

```
(c) Microsoft Corporation. Todos los derechos reservados.

D:\FlutterRemoto>git clone https://github.com/KhleiwinMPcv/ProyectoMVC_khleiwin.git
Cloning into 'ProyectoMVC_khleiwin'...
remote: Enumerating objects: 317, done.
remote: Counting objects: 100% (317/317), done.
remote: Compressing objects: 100% (251/251), done.
remote: Total 317 (delta 49), reused 223 (delta 30), pack-reused 0 (from 0)
Receiving objects: 100% (317/317), 2.08 MiB | 3.23 MiB/s, done.
Resolving deltas: 100% (49/49), done.

D:\FlutterRemoto>
```

## Sección 4: Gestión del repositorio para el proyecto

### Paso 1: Crear una Cuenta en GitHub

1. Ve a [GitHub](#).
2. Haz clic en **Sign Up**.
3. Completa el formulario con tu nombre, correo electrónico y contraseña.
4. Verifica tu correo electrónico para activar tu cuenta.

Yo omitiré este paso porque ya tengo una cuenta:

Aquí tienes el paso a paso final para **subir tu proyecto de código abierto a GitHub** de manera clara y estructurada:

### Paso 2: Crear un Repositorio en GitHub

1. Inicia sesión en tu cuenta de **GitHub**.
2. En la página principal de tu cuenta, haz clic en el ícono de tu perfil (arriba a la derecha) y selecciona **Your repositories**.

3. Haz clic en el botón **New** (Nuevo repositorio).

The screenshot shows a GitHub user profile for 'KhleiwinMPcv'. The top navigation bar includes 'Overview', 'Repositories 2', 'Projects', 'Packages', and 'Stars'. Below the navigation is a search bar with placeholder 'Find a repository...', and dropdown menus for 'Type', 'Language', and 'Sort'. A prominent green 'New' button is located on the right. Two repositories are listed: 'ProyectoMVC\_khleiwin' (Public, Dart, updated 9 minutes ago) and 'MVC\_Loguin' (Private, Python, updated 2 weeks ago). At the bottom left is an 'Edit profile' button, and at the bottom right is a copyright notice: '© 2024 GitHub, Inc.'

4. Rellena los campos de tu repositorio:

- **Repository name:** El nombre que tendrá tu repositorio (por ejemplo, mi-proyecto-open-source).
- **Description:** Breve descripción de lo que hace tu proyecto.
- **Public:** Selecciona **Public** para que tu repositorio sea accesible a todos.
- **Initialize this repository with a README:** Deja esta opción **desmarcada** si ya tienes tu proyecto listo para subir (si no, puedes marcarla y GitHub creará un archivo README).

5. Haz clic en **Create repository**.

Para acceder de forma rápida, coloca los marcadores aquí en la barra de marcadores [Importar marcadores ahora...](#) | 

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*



Repository name \*

Intranet

Intranet is available.

Great repository names are short and memorable. Need inspiration? How about [congenial-potato](#) ?

Description (optional)

htdocs y flutter para aplicación móvil Android

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

You are creating a public repository in your personal account.

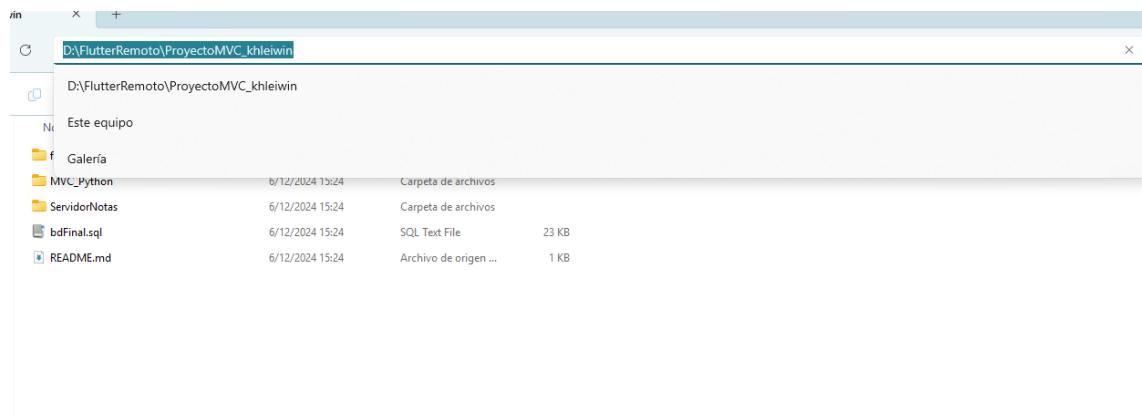
**Create repository**

The screenshot shows a GitHub repository page for 'Intranet'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and more. Below the navigation bar, the repository name 'Intranet' is displayed with a 'Public' status. To the right of the repository name are buttons for Pin, Unwatch (1), Fork (0), Star (0), and more. The main content area shows the repository's structure: 'main' branch, README files, and a detailed description of the project. The description includes the URL <https://github.com/KhleiwinMPcv/Intranet.git>, a 'Clone' button, and options to Open with GitHub Desktop or Download ZIP. On the right side, there are sections for About (htdocs y flutter para aplicación móvil Android), Readme, Activity, Statistics (0 stars, 1 watching, 0 forks), Releases (No releases published, Create a new release), and Packages (No packages published, Publish your first package). At the bottom, there are links for Terms, Privacy, Security, Status, Docs, Contact, Manage cookies, and Do not share my personal information. The GitHub logo and copyright information are also present.

## Paso 3: Subir Tu Proyecto a GitHub

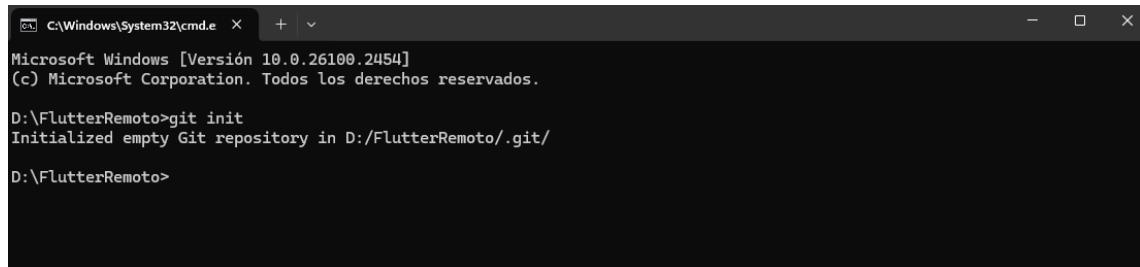
### 3.1. Preparar tu Proyecto Local

1. Abre tu terminal o PowerShell.
2. Navega hasta el directorio donde está tu proyecto



### 3.2. Inicializar un Repositorio Git Local

1. Si tu proyecto no es aún un repositorio Git, inicializa uno



```
C:\Windows\System32\cmd.exe + - Microsoft Windows [Versión 10.0.26100.2454] (c) Microsoft Corporation. Todos los derechos reservados. D:\FlutterRemoto>git init Initialized empty Git repository in D:/FlutterRemoto/.git/ D:\FlutterRemoto>
```

### 3.3. Configurar el Repositorio Remoto

1. Conecta tu repositorio local con el repositorio de GitHub usando la URL de tu repositorio (la puedes copiar desde la página de tu repositorio en GitHub):

```
D:\FlutterRemoto>git remote add origin https://github.com/KhleiwinMPcv/Intranet.git
```

### 3.4. Agregar los Archivos y Realizar un Commit

1. Agrega todos los archivos del proyecto al área de preparación:

```
D:\FlutterRemoto>git add .  
warning: in the working copy of 'flutter_application_notas_remoto/.metadata', LF will be replaced by CRLF the next time  
Git touches it  
warning: in the working copy of 'flutter_application_notas_remoto/linux/flutter/generated_plugin_registrant.cc', LF will  
be replaced by CRLF the next time Git touches it  
warning: in the working copy of 'flutter_application_notas_remoto/linux/flutter/generated_plugin_registrant.h', LF will  
be replaced by CRLF the next time Git touches it  
warning: in the working copy of 'flutter_application_notas_remoto/linux/flutter/generated_plugins.cmake', LF will be rep  
laced by CRLF the next time Git touches it  
warning: in the working copy of 'flutter_application_notas_remoto/macos/Flutter/GeneratedPluginRegistrant.swift', LF wil  
l be replaced by CRLF the next time Git touches it  
warning: in the working copy of 'flutter_application_notas_remoto/windows/flutter/generated_plugin_registrant.cc', LF will  
be replaced by CRLF the next time Git touches it  
warning: in the working copy of 'flutter_application_notas_remoto/windows/flutter/generated_plugin_registrant.h', LF wil  
l be replaced by CRLF the next time Git touches it  
warning: in the working copy of 'flutter_application_notas_remoto/windows/flutter/generated_plugins.cmake', LF will be r  
eplaced by CRLF the next time Git touches it
```

2. Realiza un commit de esos archivos:

```

D:\FlutterRemoto>git commit -m "Primer commit de mi proyecto de código abierto de Intranet"
[master (root-commit) f9248b7] Primer commit de mi proyecto de código abierto de Intranet
163 files changed, 6087 insertions(+)
create mode 100644 ServidorNotas0/Conexion.php
create mode 100644 ServidorNotas0/Conexion1.php
create mode 100644 ServidorNotas0/Controla.php
create mode 100644 ServidorNotas0/Controlal.php
create mode 100644 ServidorNotas0/Negocio.php
create mode 100644 ServidorNotas0/Negociol.php
create mode 100644 ServidorNotas0/PryNota.rar
create mode 100644 ServidorNotas0/fotos/A0001.jpg
create mode 100644 ServidorNotas0/fotos/A0002.jpg
create mode 100644 ServidorNotas0/fotos/A0003.jpg
create mode 100644 ServidorNotas0/fotos/A0004.jpg
create mode 100644 ServidorNotas0/fotos/A0005.jpg
create mode 100644 ServidorNotas0/fotos/A0006.jpg
create mode 100644 ServidorNotas0/fotos/A0007.jpg
create mode 100644 ServidorNotas0/fotos/A0008.jpg
create mode 100644 ServidorNotas0/fotos/A0011.jpg
create mode 100644 ServidorNotas0/fotos/A0012.jpg
create mode 100644 ServidorNotas0/fotos/A0013.jpg
create mode 100644 ServidorNotas0/fotos/thumbs.db
create mode 100644 ServidorNotas0/fotos/sin_foto.jpg
create mode 100644 ServidorNotas0/index.php
create mode 100644 ServidorNotas0/nbproject/private/config.properties
create mode 100644 ServidorNotas0/nbproject/private/private.properties
create mode 100644 ServidorNotas0/nbproject/private/private.xml
create mode 100644 ServidorNotas0/nbproject/project.properties
create mode 100644 ServidorNotas0/nbproject/project.xml
create mode 100644 ServidorNotas0/suma.php
create mode 100644 flutter_application_notas_remoto/.gitignore
create mode 100644 flutter_application_notas_remoto/.metadata
create mode 100644 flutter_application_notas_remoto/README.md
create mode 100644 flutter_application_notas_remoto/analysis_options.yaml
create mode 100644 flutter_application_notas_remoto/android/.gitignore
create mode 100644 flutter_application_notas_remoto/android/app/build.gradle
create mode 100644 flutter_application_notas_remoto/android/app/src/debug/AndroidManifest.xml
create mode 100644 flutter_application_notas_remoto/android/app/src/main/AndroidManifest.xml
create mode 100644 flutter_application_notas_remoto/android/app/src/main/kotlin/com/example/flutter_application_notas_remoto/MainActivity.kt
create mode 100644 flutter_application_notas_remoto/android/app/src/main/res/drawable-v21/launch_background.xml
create mode 100644 flutter_application_notas_remoto/android/app/src/main/res/drawable/launch_background.xml
create mode 100644 flutter_application_notas_remoto/android/app/src/main/res/mipmap-hdpi/ic_launcher.png
create mode 100644 flutter_application_notas_remoto/android/app/src/main/res/mipmap-mdpi/ic_launcher.png
create mode 100644 flutter_application_notas_remoto/android/app/src/main/res/mipmap-xhdpi/ic_launcher.png
create mode 100644 flutter_application_notas_remoto/android/app/src/main/res/mipmap-xxhdpi/ic_launcher.png
create mode 100644 flutter_application_notas_remoto/android/app/src/main/res/mipmap-xxxhdpi/ic_launcher.png

create mode 100644 flutter_application_notas_remoto/ios/Runner.xcodeproj/project.xcworkspace/xcscheme/xcschemes/Runner.xcscheme
create mode 100644 flutter_application_notas_remoto/ios/Runner.xcodeproj/xcsshareddata/xcschemas/Runner.xcscheme
create mode 100644 flutter_application_notas_remoto/ios/Runner.xcworkspace/contents.xcworkspacepdata
create mode 100644 flutter_application_notas_remoto/ios/Runner.xcworkspace/xcshareddata/IDEWorkspaceChecks.plist
create mode 100644 flutter_application_notas_remoto/ios/Runner.xcworkspace/xcshareddata/WorkspaceSettings.xcssettings
create mode 100644 flutter_application_notas_remoto/ios/Runner/AppDelegate.swift
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Contents.json
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-1024x1024@1x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-20x20@1x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-20x20@2x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-20x20@3x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-29x29@1x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-29x29@2x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-29x29@3x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-40x40@1x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-40x40@2x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-40x40@3x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-60x60@2x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-60x60@3x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-76x76@1x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-76x76@2x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/AppIcon.appiconset/Icon-App-83.5x83.5@2x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/LaunchImage.imageset/Contents.json
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/LaunchImage.imageset/LaunchImage.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/LaunchImage.imageset/LaunchImage@2x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Assets.xcassets/LaunchImage.imageset/LaunchImage@3x.png
create mode 100644 flutter_application_notas_remoto/ios/Runner/Base.lproj/LaunchScreen.storyboard
create mode 100644 flutter_application_notas_remoto/ios/Runner/Base.lproj/Main storyboard
create mode 100644 flutter_application_notas_remoto/ios/Runner/Info.plist
create mode 100644 flutter_application_notas_remoto/ios/Runner/Bridging-Header.h
create mode 100644 flutter_application_notas_remoto/ios/RunnerTests/RunnerTests.swift
create mode 100644 flutter_application_notas_remoto/lib/clases/alumno.dart
create mode 100644 flutter_application_notas_remoto/lib/clases/curso.dart
create mode 100644 flutter_application_notas_remoto/lib/clases/menu.dart
create mode 100644 flutter_application_notas_remoto/lib/clases/nota.dart
create mode 100644 flutter_application_notas_remoto/lib/clases/pago.dart
create mode 100644 flutter_application_notas_remoto/lib/cnotas.dart
create mode 100644 flutter_application_notas_remoto/lib/pagnota.dart
create mode 100644 flutter_application_notas_remoto/lib/pagnotaxalumno.dart
create mode 100644 flutter_application_notas_remoto/lib/pagpagosxalumno.dart
create mode 100644 flutter_application_notas_remoto/linux/.gitignore
create mode 100644 flutter_application_notas_remoto/linux/CMakeLists.txt
create mode 100644 flutter_application_notas_remoto/linux/Flutter/CMakeLists.txt
create mode 100644 flutter_application_notas_remoto/linux/flutter/generated_plugin_registrant.cc
create mode 100644 flutter_application_notas_remoto/linux/flutter/generated_plugin_registrant.h
create mode 100644 flutter_application_notas_remoto/linux/flutter/generated_plugins.cmake
create mode 100644 flutter_application_notas_remoto/linux/main.cc
create mode 100644 flutter_application_notas_remoto/linux/my_application.cc
create mode 100644 flutter_application_notas_remoto/linux/my_application.h
create mode 100644 flutter_application_notas_remoto/macos/.gitignore

```

```
create mode 100644 flutter_application_notas_remoto/macos/Runner/Configs/AppInfo.xcconfig
create mode 100644 flutter_application_notas_remoto/macos/Runner/Configs/Debug.xcconfig
create mode 100644 flutter_application_notas_remoto/macos/Runner/Configs/Release.xcconfig
create mode 100644 flutter_application_notas_remoto/macos/Runner/Configs/Warnings.xcconfig
create mode 100644 flutter_application_notas_remoto/macos/Runner/DebugProfile.entitlements
create mode 100644 flutter_application_notas_remoto/macos/Runner/Info.plist
create mode 100644 flutter_application_notas_remoto/macos/Runner/MainFlutterWindow.swift
create mode 100644 flutter_application_notas_remoto/macos/Runner/Release.entitlements
create mode 100644 flutter_application_notas_remoto/macos/RunnerTests/RunnerTests.swift
create mode 100644 flutter_application_notas_remoto/pubspec.lock
create mode 100644 flutter_application_notas_remoto/pubspec.yaml
create mode 100644 flutter_application_notas_remoto/web/favicon.png
create mode 100644 flutter_application_notas_remoto/web/icons/Icon-192.png
create mode 100644 flutter_application_notas_remoto/web/icons/Icon-512.png
create mode 100644 flutter_application_notas_remoto/web/icons/Icon-maskable-192.png
create mode 100644 flutter_application_notas_remoto/web/icons/Icon-maskable-512.png
create mode 100644 flutter_application_notas_remoto/web/index.html
create mode 100644 flutter_application_notas_remoto/web/manifest.json
create mode 100644 flutter_application_notas_remoto/windows/.gitignore
create mode 100644 flutter_application_notas_remoto/windows/CMakeLists.txt
create mode 100644 flutter_application_notas_remoto/windows/flutter/CMakeLists.txt
create mode 100644 flutter_application_notas_remoto/windows/flutter/generated_plugin_registrant.cc
create mode 100644 flutter_application_notas_remoto/windows/flutter/generated_plugin_registrant.h
create mode 100644 flutter_application_notas_remoto/windows/flutter/generated_plugins.cmake
create mode 100644 flutter_application_notas_remoto/windows/runner/CMakeLists.txt
create mode 100644 flutter_application_notas_remoto/windows/runner/Runner.rc
create mode 100644 flutter_application_notas_remoto/windows/runner/flutter_window.cpp
create mode 100644 flutter_application_notas_remoto/windows/runner/flutter_window.h
create mode 100644 flutter_application_notas_remoto/windows/runner/main.cpp
create mode 100644 flutter_application_notas_remoto/windows/runner/resource.h
create mode 100644 flutter_application_notas_remoto/windows/runner/resources/app_icon.ico
create mode 100644 flutter_application_notas_remoto/windows/runner/runner.exe.manifest
create mode 100644 flutter_application_notas_remoto/windows/runner/utils.cpp
create mode 100644 flutter_application_notas_remoto/windows/runner/utils.h
create mode 100644 flutter_application_notas_remoto/windows/runner/win32_window.cpp
create mode 100644 flutter_application_notas_remoto/windows/runner/win32_window.h
```

D:\FlutterRemoto>

### 3.5. Subir los Archivos a GitHub

#### 1. Subir el proyecto al repositorio de GitHub

**Nota: Si tu repositorio usa main en lugar de master como nombre de la rama principal, usa main:**

**git push -u origin main**

```
D:\FlutterRemoto>git push -u origin master
Enumerating objects: 217, done.
Counting objects: 100% (217/217), done.
Delta compression using up to 16 threads
Compressing objects: 100% (186/186), done.
Writing objects: 100% (217/217), 377.54 KiB | 7.87 MiB/s, done.
Total 217 (delta 22), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (22/22), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/KhleiwinMPcv/Intranet/pull/new/master
remote:
To https://github.com/KhleiwinMPcv/Intranet.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

D:\FlutterRemoto>

Este es el resultado:

The screenshot shows a GitHub repository page for a project named 'Intranet'. At the top, there's a yellow banner indicating 'master had recent pushes 3 seconds ago'. Below the banner, there are buttons for 'Pin', 'Unwatch', 'Fork', 'Star', and a 'Compare & pull request' button. The repository has one commit from 'KhleiwinMPcv' labeled 'Initial commit' made 20 minutes ago. A file named 'README.md' is listed with an 'Initial commit' and a timestamp of '20 minutes ago'. On the right side, there are sections for 'About', 'Readme', 'Activity', 'Stars', 'Watching', 'Forks', 'Releases' (which says 'No releases published'), and 'Packages' (which says 'No packages published').

#### Paso 4: Configurar una Licencia de Código Abierto (Opcional)

1. En tu repositorio de GitHub, ve a Add file y selecciona Create new file.
2. Nombra el archivo como LICENSE.
3. Escoge una licencia de código abierto. Por ejemplo, puedes usar la MIT License:
  - o Copia el texto de la MIT License y pégalo en el archivo LICENSE.
4. Haz clic en Commit changes...

```
1  MIT License
2
3  Copyright (c) [2024] [Khleiwin]
4
5  Se concede permiso, de forma gratuita, a cualquier persona que obtenga una copia de este software y
6  los archivos de documentación asociados (Intranet: FluterRemoto), para usar el Software sin restricciones,
7  incluyendo sin limitación los derechos para usar, copiar, modificar, fusionar, publicar, distribuir,
8  sublicenciar y/o vender copias del Software, y para permitir a las personas a quienes se les proporcione el
9  Software hacerlo, sujeto a las siguientes condiciones:
10
11 La mencionada copia del Software debe incluir el aviso de copyright anterior,
12 esta lista de condiciones y la siguiente exención de responsabilidad.
13
14 EL SOFTWARE SE PROPORCIONA "TAL CUAL", SIN GARANTÍAS DE NINGÚN TIPO,
15 EXPRESAS O IMPLÍCITAS, INCLUYENDO PERO NO LIMITADO A LAS GARANTÍAS DE COMERCIABILIDAD,
16 APTITUD PARA UN PROPÓSITO PARTICULAR Y NO INFRACCIÓN. EN NINGÚN CASO LOS AUTORES O
17 LOS TITULARES DEL COPYRIGHT SERÁN RESPONSABLES POR CUALQUIER RECLAMO, DAÑOS O CUALQUIER OTRA RESPONSABILIDAD,
18 YA SEA EN UNA ACCIÓN DE CONTRATO, AGRAVIO O CUALQUIER OTRA ACCIÓN, QUE SURJA DE O EN CONEXIÓN
19 CON EL SOFTWARE O EL USO U OTROS TRATOS EN EL SOFTWARE.
20
```

Se ve así el actualizado:

The screenshot shows a GitHub repository interface. At the top, a commit from user 'KhleiwinMPcv' titled 'Create LICENCE' is shown, made 1 minute ago. Below it, two files are listed: 'LICENCE' and 'README.md'. 'LICENCE' was created 1 minute ago, and 'README.md' was an initial commit 40 minutes ago. A red horizontal bar highlights the 'README' and 'License' sections. In the main content area, another commit from 'KhleiwinMPcv' is shown, stating 'This branch is 2 commits ahead of, 2 commits behind main.' A 'Contribute' button is visible. A table lists the repository's contents: 'ServidorNotasO', 'flutter\_application\_notas\_remoto', and 'LICENCE'. The 'LICENCE' file is the most recent commit, made 'now'.

Name	Last commit message	Last commit date
ServidorNotasO	Primer commit de mi proyecto de código abierto de Intra...	25 minutes ago
flutter_application_notas_remoto	Primer commit de mi proyecto de código abierto de Intra...	25 minutes ago
LICENCE	Create LICENCE	now

## Paso 5: Agregar un Archivo de Contribución

1. En tu repositorio de GitHub, haz clic en **Add file** y selecciona **Create new file**.
2. Nombra el archivo como **CONTRIBUTING.md**.
3. Agrega información sobre cómo las personas pueden contribuir a tu proyecto, por ejemplo

KhleiwinMPcv / Intranet

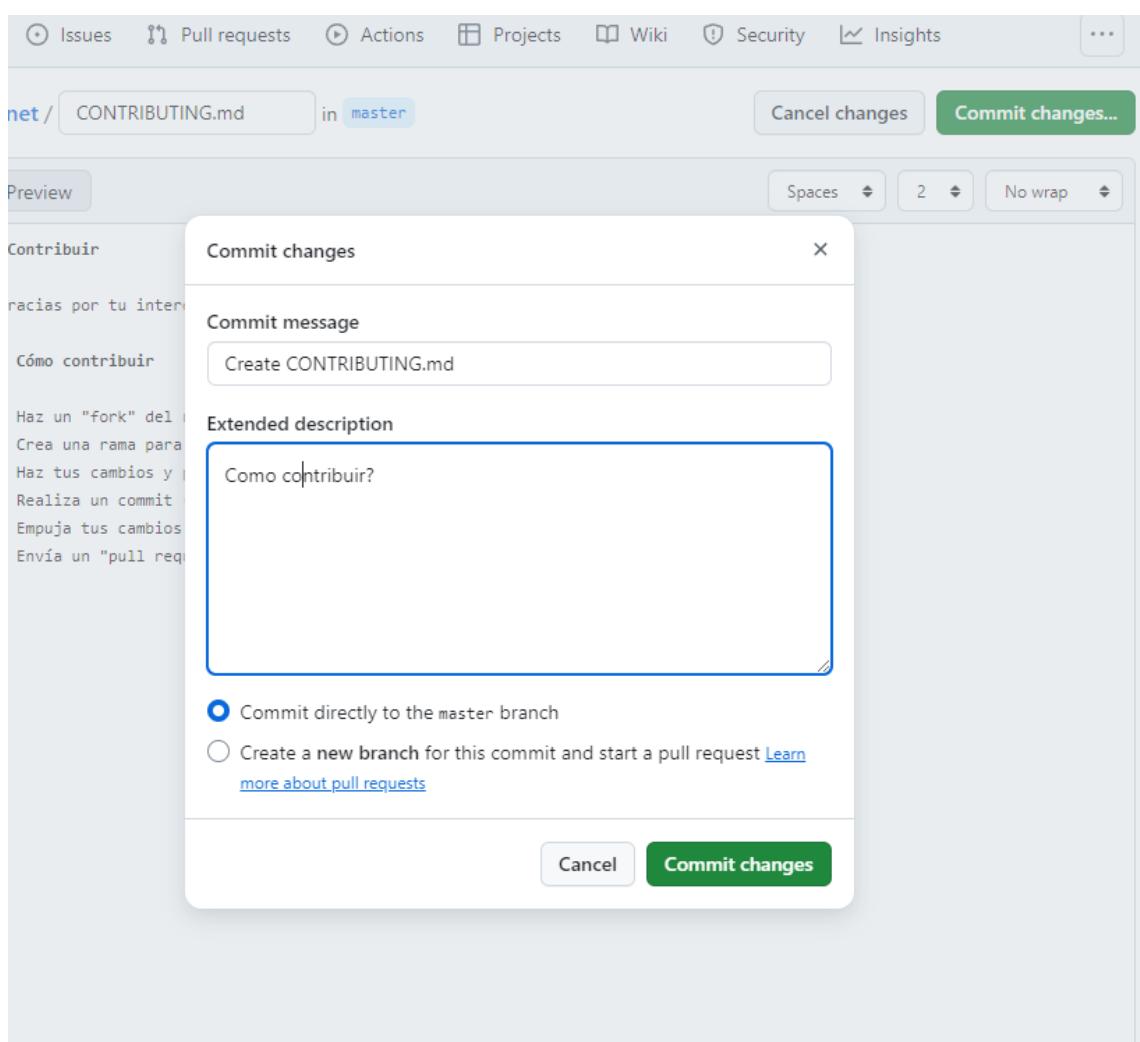
Code Issues Pull requests Actions Projects Wiki Security Insights

Intranet / CONTRIBUTING.md in master

Cancel changes Commit changes...

Edit Preview Spaces 2 No wrap

```
1 # Contribuir
2
3 ¡Gracias por tu interés en contribuir a este proyecto!
4
5 ## Cómo contribuir
6
7 1. Haz un "fork" del repositorio.
8 2. Crea una rama para tu característica (`git checkout -b feature/nueva-caracteristica`).
9 3. Haz tus cambios y prueba que todo funcione correctamente.
10 4. Realiza un commit (`git commit -am 'Agrega nueva característica'`).
11 5. Empuja tus cambios a tu repositorio (`git push origin feature/nueva-caracteristica`).
12 6. Envía un "pull request" desde tu rama al repositorio original.
13
```



## Actualizado:

The screenshot shows a GitHub repository page for 'Intranet' under the user 'KhleiwinMPcv'. The 'Code' tab is selected. The current branch is 'master', which has 3 commits ahead of and 2 commits behind the 'main' branch. The last commit was made by 'KhleiwinMPcv' at 'e8fb941 · now'. A 'Contribute' button is visible. Below the commit list, there is a table showing file details.

Name	Last commit message	Last commit date
ServidorNotasO	Primer commit de mi proyecto de código abierto de Intra...	31 minutes ago
flutter_application_notas_remoto	Primer commit de mi proyecto de código abierto de Intra...	31 minutes ago
CONTRIBUTING.md	Create CONTRIBUTING.md	now
LICENCE	Create LICENCE	5 minutes ago