



**UNIVERSIDAD TECNOLÓGICA DE EL SALVADOR
FACULTAD DE INFORMÁTICA
Y CIENCIAS APLICADAS
ESCUELA DE INFORMÁTICA
PROGRAMACIÓN III, SECCIÓN 01**

DOCENTE:

JORGE ALBERTO ACEVEDO DÍAZ

ESTUDIANTES:

CARNET:

ALVARADO FLORES, DANIEL ALEJANDRO	25-1664-2020
ASCENCIO ZÚNIGA, MADELINE NICOLE	17-3762-2020
CUATRO RIVERA, FERNANDO MIGUEL	25-0871-2020
ECHEVERRÍA SANTACRUZ, DIEGO ARMANDO	25-0061-2020
HERNÁNDEZ ROLIN, LIZBETH JAZMIN	17-0829-2020
LOPEZ ANDRES, JOSE TRANSITO	17-2444-2020
MARTÍNEZ CORTEZ, CARLOS GUILLERMO	17-1543-2020
MENJIVAR MARROQUIN, ROGELIO ISAÍ	25-2786-2020
MONTANO MUÑOZ, ERIKA ROXANA	27-2005-2020
SIGUENZA ORELLANA, CHRISTIAN JEFFRY	27-0034-2020
VÁSQUEZ CASTELLANOS, CARLOS RODOLFO	25-0828-2019

FECHA DE ENTREGA: LUNES 30 DE MAYO DE 2022

Índice

Introducción.....	4
Problema Investigado	5
Justificación.....	6
Objetivos	7
Objetivo general	7
Objetivos específicos.....	7
Diseño de la Solución.....	8
Base de datos del proyecto	8
Procedimientos almacenados de la tabla alumno	8
Tabla de contactos	10
Procedimientos almacenados de la tabla contacto.....	11
Tabla de notas.....	12
Procedimientos almacenados de la tabla notas.....	12
Procedimientos almacenados para consultar notas y contactos.	13
Validaciones de los formularios	17
¿Cómo generar el archivo .jar del proyecto?.....	17
¿Cómo validar que los formularios no se repitan al abrirlos?	18
Formularios del proyecto.....	19
Login	19

Registro20

Home20

Contactos21

Notas.....21

Conclusiones22

Recomendaciones23

Tabla de Ilustraciones.....24

Introducción

En el presente trabajo, se dejará constancia de cada uno de los pasos realizados para llevar a cabo nuestro proyecto, utilizando como lenguaje de programación Java y MySQL como lenguaje administrador de bases de datos. Dejando también evidencia sobre los respectivos diagramas que se utilizaron de base en cada paso a realizar en el proyecto.

En el trabajo, daremos a conocer sobre cómo está estructurado nuestro proyecto, implementando métodos eficaces para utilizar procedimientos almacenados, además, utilizaremos las mejores prácticas para dejar constancia de lo aprendido, también conoceremos sobre las diferentes formas o métodos que se utilizan para establecer una conexión a una base de datos utilizando el lenguaje de programación Java.

Problema Investigado

Los alumnos de la universidad se muestran frente a una complejidad real al instante de realizar sus ocupaciones, tener resoluciones tecnológicas con la suficiente liviandad y efectividad a nivel privado en la organización; es una necesidad para lograr laborar ideas sin necesidad de tener que recurrir a plataformas externas generalmente que, en lugar de contribuir al enriquecimiento de los estudiantes, los distrae. La sociedad universitaria requiere de una plataforma donde logre guardar sus notas y sus contactos con mucha confianza; una plataforma con una interfaz amistosa donde la custodia de sus notas donde sea la máxima prioridad. Frente a lo propuesto, se muestra una solución al problema, para poder mejorar el rendimiento universitario a modo de un bloc personal de escrito plano con sus respectivos usuarios.

Justificación

En un contexto social en el que la movilidad y la globalización son hechos cotidianos, cada vez, más se hace necesario pensar en herramientas que han sido útiles a lo largo de los años para que lo sigan siendo en este nuevo escenario en donde los puestos de trabajo, estudio o de ocio están allí, donde está esa conexión de internet, enfocándose en la idea donde la tecnología ha de servir para simplificar la vida cotidiana, con el tiempo se ha visto la necesidad de crear una herramienta tecnológica en el cual facilite al campo estudiantil logrando minimizar tiempo y ayudar a organizar y administrar actividades presentes y futuras por medio de un bloc de notas.

El presente proyecto se enfoca en la realización de un sistema en el cual consiste en mantener la simplicidad para aquellos estudiantes que solo buscan crear notas de forma rápida y sencilla, notas que pueden servir de base para otras herramientas más sofisticadas o que simplemente después de cumplir su papel son eliminadas, ofrece al campo estudiantil el manejo de información de actividades como: agregar apuntes y agregar contactos donde no habrá necesidad de utilizar diferentes aplicaciones ajenas a esta que muchas veces no brindan la confiabilidad necesaria.

La aportación principal de este proyecto será proveer de una vía alterna, sencilla y fiable para la simple gestión de notas personales o profesionales, brindando seguridad y confiabilidad a la información agregada a dicho bloc.

Objetivos

Objetivo general

Crear una herramienta tecnológica dirigida al sector estudiantil con el fin realizar diversas actividades como: agregar contactos, agendar actividades representadas en notas, toma de apuntes, en un espacio virtual la cual pretende tener un ambiente exclusivamente educativo con enfoque en la optimización de tiempo y mejor administración de actividades académicas.

Objetivos específicos

1. Lograr un espacio estudiantil privado y exclusivo en el que los estudiantes se sientan seguros de guardar información privada de índole educativa
2. Diseñar un sistema fácil e intuitivo con una integración a una Base de Datos segura que será utilizado para y por los usuarios dentro del sector estudiantil.
3. Desplegar soluciones tecnológicas necesarias para poder integrar en el programa deseado los distintos componentes individuales dentro de la aplicación así logrando la funcionalidad deseada aplicando lo expuesto en clase para la necesidad.

Diseño de la Solución

Base de datos del proyecto

Como bien mencionamos, se realizó la base de datos en MySQL. En total son 3 tablas y 18 procedimientos almacenados, de igual forma, se utiliza la persistencia en cascada para insertar, eliminar y actualizar en las tablas.

```
create table alumno (  
  id_alumno INT NOT NULL AUTO_INCREMENT,  
  carnet VARCHAR(25) NOT NULL,  
  nombre VARCHAR(25) NOT NULL,  
  apellido VARCHAR(25) NOT NULL,  
  carrera VARCHAR(75) NOT NULL,  
  email VARCHAR(50) NOT NULL,  
  pw VARCHAR(250) NOT NULL,  
  CONSTRAINT pk_id_alumno PRIMARY KEY(id_alumno),  
  CONSTRAINT uq_carnet UNIQUE (carnet)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
```

Ilustración A. Tabla Alumno.

Procedimientos almacenados de la tabla alumno

```
-- Procedimiento almacenado para el inicio de sesion por medio del login en la tabla alumno  
USE tecnotes_db_2;  
DROP PROCEDURE IF EXISTS PA_login_alumno;  
CREATE PROCEDURE PA_login_alumno(  
  IN carnet_1 VARCHAR(25),  
  IN pw_1 VARCHAR(250)  
) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY  
DEFINER  
SELECT * FROM alumno WHERE carnet = carnet_1 AND pw = pw_1;
```

Ilustración B. Procedimiento almacenado de Login


```

-- Procedimiento para insertar alumno en la tabla alumno
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_insertar_alumno;
CREATE PROCEDURE PA_insertar_alumno(
IN carnet    VARCHAR(25),
IN nombre    VARCHAR(25),
IN apellido  VARCHAR(25),
IN carrera   VARCHAR(75),
IN email     VARCHAR(50),
IN pw        VARCHAR(250)
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
INSERT INTO alumno VALUES (NULL,carnet,nombre,apellido,carrera,email,pw);

```

Ilustración C. Procedimientos almacenados para agregar

```

-- Procedimiento para editar alumno en la tabla alumno
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_actualizar_alumno;
CREATE PROCEDURE PA_actualizar_alumno(
IN id_alumno2 INT,
IN carrera2    VARCHAR(75),
IN email2      VARCHAR(50)
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
UPDATE alumno SET carrera=carrera2,email=email2

```

Ilustración D. Procedimiento almacenado para actualizar alumno

```
-- Procedimiento para buscar alumno en la tabla alumno
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_buscar_alumnos;
CREATE PROCEDURE PA_buscar_alumnos(
IN id_alumno1 INT
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
SELECT * FROM alumno WHERE id_alumno=id_alumno1;
```

Ilustración E. Procedimiento almacenado para buscar alumno

```
-- Procedimiento para eliminar alumno en la tabla alumno
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_eliminar_alumno;
CREATE PROCEDURE PA_eliminar_alumno(
IN id_alumno1 INT
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
DELETE FROM alumno WHERE id_alumno=id_alumno1;
```

Ilustración F. Procedimiento almacenado para eliminar alumno

Tabla de contactos

```
create table contactos (
id_contactos      INT          NOT NULL AUTO_INCREMENT,
nombre            VARCHAR(25)  NOT NULL,
apellido          VARCHAR(25)  NOT NULL,
telefono          VARCHAR(15)  NOT NULL,
email             VARCHAR(50)  NOT NULL,
carnet_contacto   VARCHAR(25)  NOT NULL,
apunte            TEXT          NOT NULL,
id_alumno         INT          NOT NULL,
CONSTRAINT pk_id_contactos PRIMARY KEY(id_contactos)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
```

Ilustración G. Tabla contactos

Procedimientos almacenados de la tabla contacto

```
-- Procedimiento almacenado para insertar en la tabla contactos
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_insertar_contactos;
CREATE PROCEDURE PA_insertar_contactos(
  IN nombre VARCHAR(25),
  IN apellido VARCHAR(25),
  IN telefono VARCHAR(15),
  IN email VARCHAR(50),
  IN carnet_contacto VARCHAR(25),
  IN apunte TEXT,
  IN id_alumno INT
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
INSERT INTO contactos VALUES (NULL,nombre,apellido,telefono,email,carnet_contacto,apunte,id_alumno);
```

Ilustración H. Procedimiento almacenado para insertar contacto

```
-- procedimiento almacenado para actualizar la tabla contactos
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_actualizar_contactos;
create procedure PA_actualizar_contactos(
  IN id_contactos2 INT,
  IN nombre2 VARCHAR(25),
  IN apellido2 VARCHAR(25),
  IN telefono2 VARCHAR(15),
  IN email2 VARCHAR(50),
  IN carnet_contacto2 VARCHAR(25),
  IN punte2 TEXT
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
UPDATE contactos SET nombre=nombre2,apellido=apellido2,telefono=telefono2,email=email2,
carnet_contacto=carnet_contacto2, apunte=punte2
WHERE id_contactos=id_contactos2;
```

Ilustración I. Procedimiento almacenado para actualizar contacto

```
-- procedimiento para eliminar un contacto de la tabla contactos
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_eliminar_contactos;
CREATE PROCEDURE PA_eliminar_contactos(
IN id_contactos1 INT
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
DELETE FROM contactos WHERE id_contactos=id_contactos1;
```

Ilustración J. Procedimiento almacenado para eliminar contacto

Tabla de notas

```
create table notas (
id_notas INT NOT NULL AUTO_INCREMENT,
titulo VARCHAR(25) NOT NULL,
notas TEXT NOT NULL,
id_alumno INT NOT NULL,
CONSTRAINT pk_id_notas PRIMARY KEY(id_notas)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
```

Ilustración K. Tabla de notas

Procedimientos almacenados de la tabla notas

```
-- Procedimiento almacenado para insertar en la tabla notas
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_insertar_notas;
create procedure PA_insertar_notas(
IN titulo VARCHAR(25),
IN notas TEXT,
IN id_alumno INT
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
INSERT INTO notas VALUES(NULL,titulo,notas,id_alumno);
```

Ilustración L. Procedimiento almacenado para insertar nota

```

-- procedimiento almacenado para actualizar la tabla contactos
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_actualizar_notas;
CREATE PROCEDURE PA_actualizar_notas(
  IN id_notas2 INT,
  IN titulo2 VARCHAR(25),
  IN notas2 TEXT
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
UPDATE notas SET titulo=titulo2,notas=notas2
WHERE id_notas=id_notas2;

```

Ilustración M. Procedimiento almacenado para actualizar nota

```

-- procedimiento para eliminar la tabla una nota de la tabla notas
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_eliminar_notas;
CREATE PROCEDURE PA_eliminar_notas(
  id_notas1 INT
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
DELETE FROM notas WHERE id_notas=id_notas1;

```

Ilustración N. Procedimiento almacenado para eliminar una nota

Procedimientos almacenados para consultar notas y contactos.

```

-- Consulta para el alumno a sus contactos
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_consulta_contactos;
CREATE PROCEDURE PA_consulta_contactos (
  IN id_alumno1 INT
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
SELECT * FROM contactos WHERE id_alumno = id_alumno1 ORDER BY id_contactos DESC;

```

Ilustración O. Procedimiento almacenado para consultar los contactos de alumno

```
-- Consulta para el alumno
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_consulta_notas;
CREATE PROCEDURE PA_consulta_notas(
IN id_alumno1 INT
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
SELECT * FROM notas WHERE id_alumno = id_alumno1 ORDER BY id_notas DESC;
```

Ilustración P. Procedimiento almacenado para consultar las notas de alumno

```
-- Consulta para el alumno
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_cantidad_contactos;
CREATE PROCEDURE PA_cantidad_contactos(
IN id_alumno1 INT
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
SELECT COUNT(*) AS cantidad_contactos FROM contactos WHERE id_alumno = id_alumno1;
```

Ilustración Q. Procedimiento almacenado para obtener la cantidad de contactos de alumno

```
-- Consulta para el alumno
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_cantidad_notas;
CREATE PROCEDURE PA_cantidad_notas(
IN id_alumno1 int
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
SELECT COUNT(*) AS cantidad_notas FROM notas WHERE id_alumno = id_alumno1;
```

Ilustración R. Procedimiento almacenado para obtener la cantidad de notas de alumno

```
-- Procedimiento obtener la cantidad de carnet existentes
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_global_carnet;
CREATE PROCEDURE PA_global_carnet(
IN carnet_1 VARCHAR(25)
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
SELECT COUNT(*) AS cantidad_carnet FROM alumno WHERE carnet = carnet_1;
```

Ilustración S. Procedimiento almacenado para obtener la cantidad de carnet existentes

```
-- Procedimiento obtener la cantidad de carnet existentes
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_global_email;
CREATE PROCEDURE PA_global_email(
IN email_1 VARCHAR(150)
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
SELECT COUNT(*) AS cantidad_email FROM alumno WHERE email = email_1;
```

Ilustración T. Procedimiento almacenado para obtener la cantidad de emails existentes

```
-- Procedimiento para procesar la nota automatica al registrarse
USE tecnotes_db_2;
DROP PROCEDURE IF EXISTS PA_nota_defecto;
create procedure PA_nota_defecto(
IN nombre VARCHAR(50),
IN id_alumno INT
)NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER
INSERT INTO notas VALUES(NULL,"¡Hola, gracias!", concat("Gracias por registrarte ", nombre ,", disfruta de TecNotes."), id_alumno);
```

Ilustración U. Procedimientos almacenados para procesar una nota de manera automática al estar registrado

Diagrama de la base de datos:

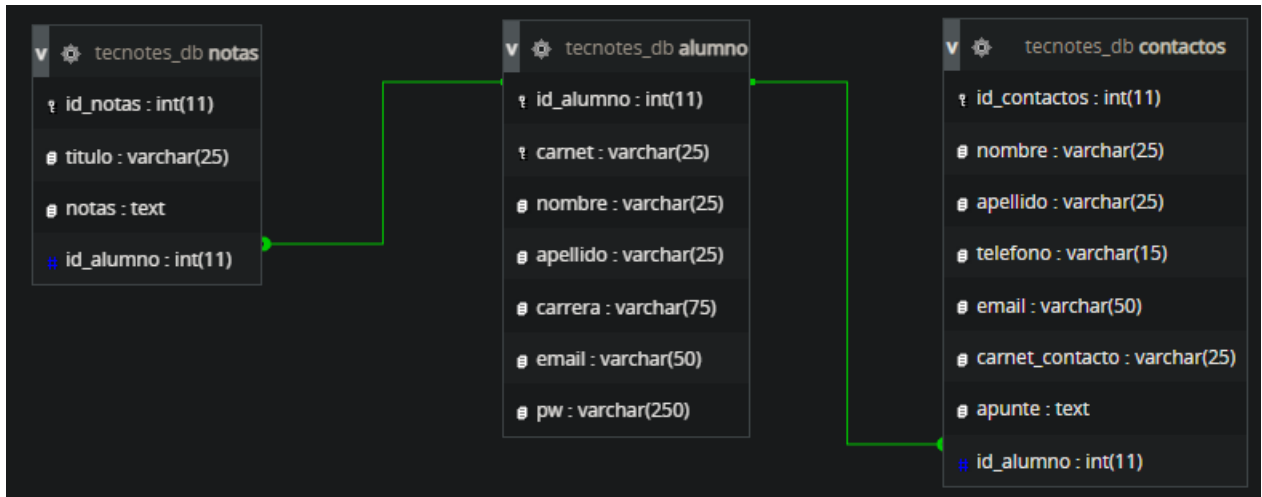


Ilustración V. Diagrama de la base de datos generado por MySQL

Diagrama entidad-relación:

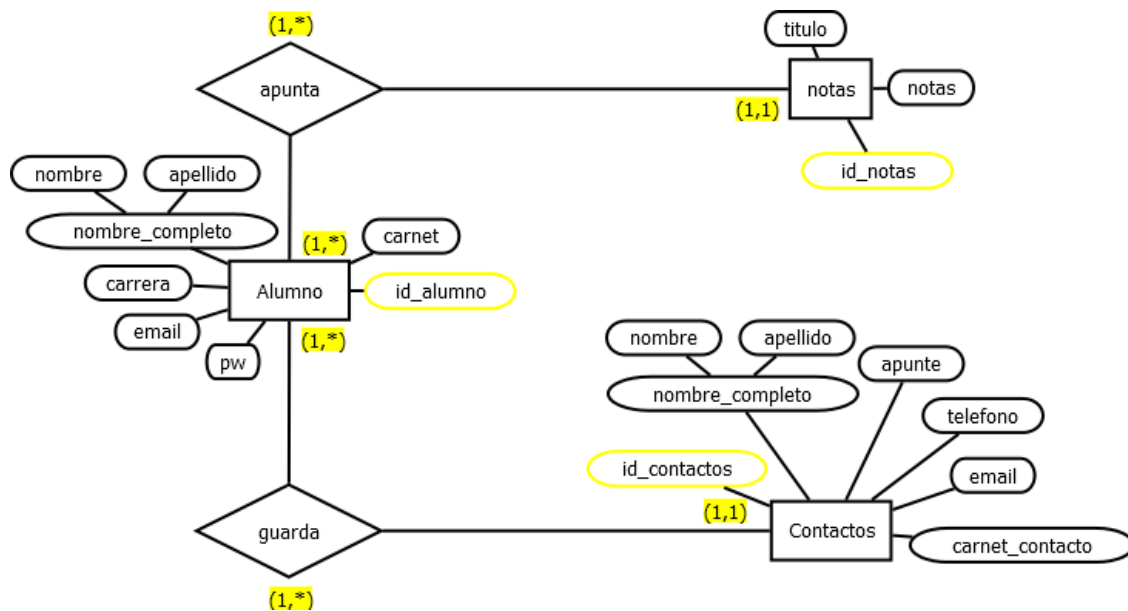


Ilustración 19 Diagrama Entidad-Relación

Diagrama relacional:

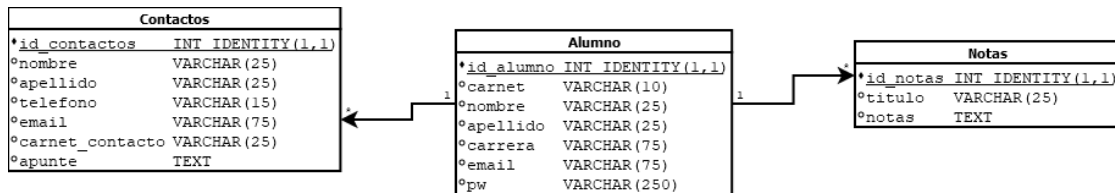


Ilustración 20. Diagrama Relacional

Validaciones de los formularios

Las validaciones de los formularios en NetBeans fue una de las peticiones del proyecto por parte del docente entre ellas están:

¿Cómo generar el archivo .jar del proyecto?

Simplemente basta con darle clic derecho al proyecto en el que estemos trabajando y le damos a la opción que dice “Clean and Build”, esta acción tardará dependiendo del tamaño del proyecto y la velocidad del ordenador.

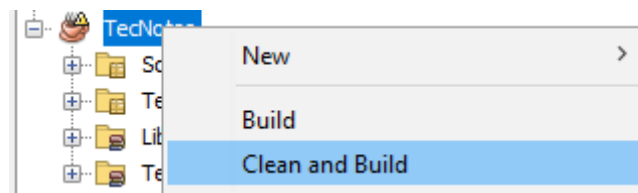


Ilustración W. Proceso del archivo .jar

Posteriormente, nos dirigimos a donde tenemos guardado nuestro proyecto, y nos movemos a la carpeta que dice “dist”, dentro de esa carpeta se encuentra nuestro archivo .jar

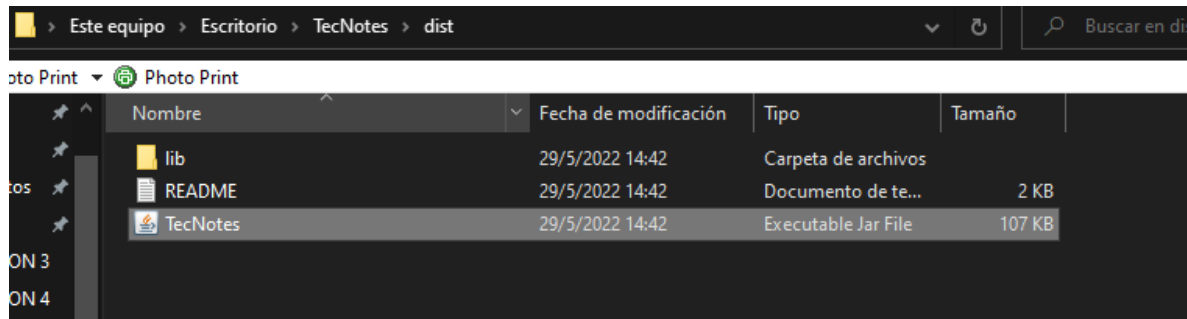


Ilustración 22 Dirección donde se guarda el archivo .jar

¿Cómo validar que los formularios no se repitan al abrirlos?

Nuestro equipo se puso de acuerdo en tratar de evadir complicaciones en código, por lo que optamos por la opción de añadir `JDialogForm` en vez de `JFrameForm`. Este formulario automáticamente hace la validación de no abrir varios formularios al presionar un botón, ya que es una ventana emergente, que no se cierra hasta cumplir su objetivo o hasta que el usuario decida salir.

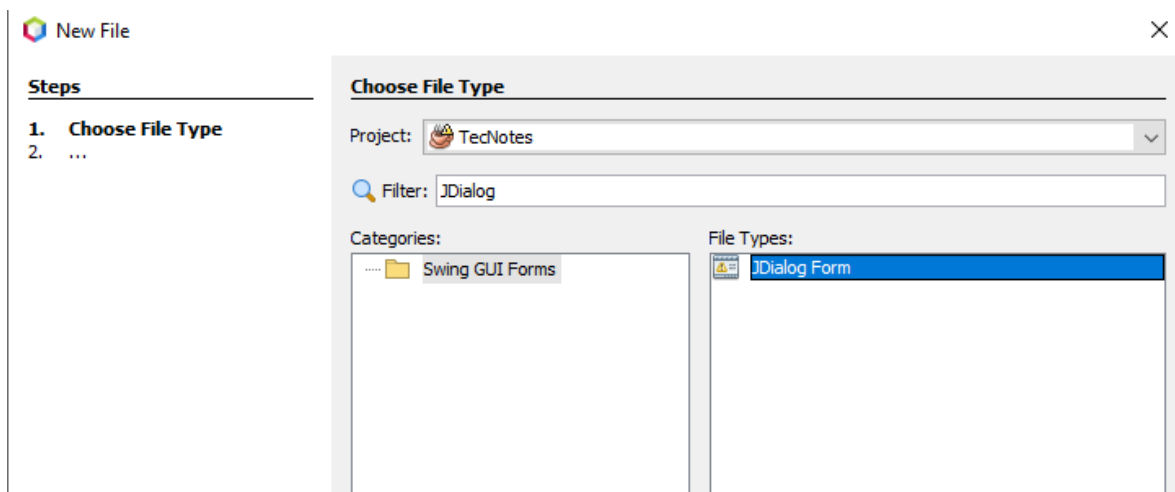


Ilustración X. `JDialogForm`

Para agregar uno de estos formularios solo basta con agregarlo. Sin embargo, si no aparece entre las opciones, nos dirigimos a la opción que dice “Other” y dentro de ahí buscamos el JDialogForm.

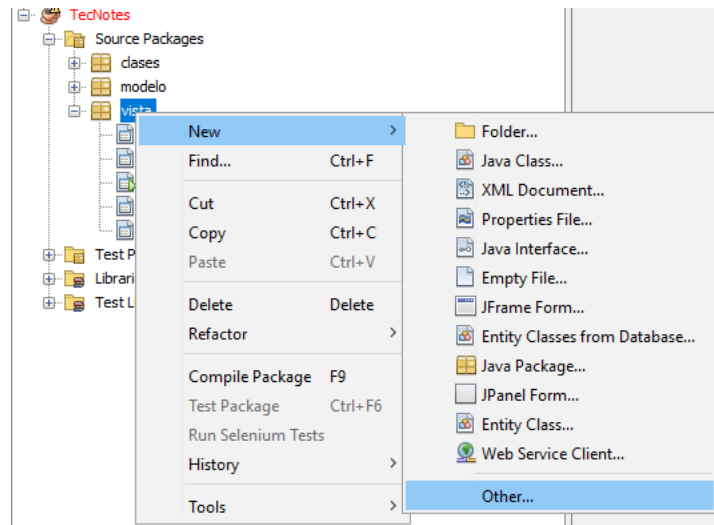


Ilustración Y. Agregar un JDialogForm

Formularios del proyecto

El proyecto se desarrolló con el lenguaje de Java, y en el IDE de NetBeans. **Todos los formularios están validados con lo mencionado anteriormente.**

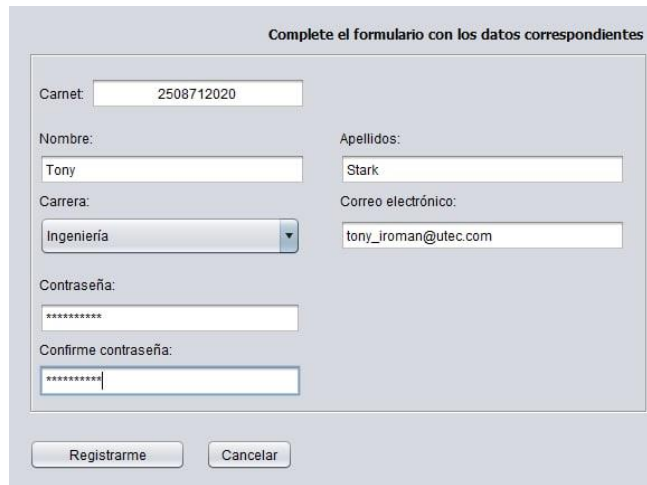
Login

A screenshot of a Java Swing window titled 'Login'. The window has a light blue background and a title bar with standard Windows controls. The main content area is titled 'Inicio de sesión' in bold black text. Below the title, there are two text input fields. The first is labeled 'Carnet:' and contains the text '2508712020'. The second is labeled 'Contraseña:' and contains a series of asterisks '*****'. Below these fields is a large, light blue button labeled 'Comenzar'. At the bottom left, there is a link that says '¿No estas registrado? [regístrate, aquí](#)'. At the bottom right, there is a smaller, light blue button labeled 'Cerrar'.

Ilustración 25 Login

Se realizo el formulario de registro para que los usuarios pudieran observar su respectiva información.

Registro



Complete el formulario con los datos correspondientes

Carnet:

Nombre: Apellidos:

Carrera: Correo electrónico:

Contraseña:

Confirme contraseña:

Ilustración 26 Registro

Posteriormente al registrarse o ya tener la cuenta, se accede al formulario de Home.

Home



Administración de contactos y notas

Contactos Notas Cerrar...

Hola Fernando Miguel, Bienvenido a tu centro de contactos y notas institucionales

Sus datos

Carnet: **2508712020**

Nombre completo: **Fernando Miguel Cuatro Rivera**

Carrera: **Técnico** Email: **fernandocuatrorivera@gmail.com**

Contactos **Notas**

2 **4**

Ilustración Z. Home

Cuando el usuario presiona el botón de “Mis contactos”, lo redirige al formulario de contactos.

Contactos

Hola, visualiza, guarda, edita y borra tus contactos

Contactos

Nombres: Apellidos:

Teléfono: (+503) Email:

Carnet del contacto: Apunte:

	Nombres	Apellidos	Teléfono	Email	Carnet	Apunte
2	Diana Ivette	Torrento Rodriguez	64000000	diana_torrento@utec.com	3583932019	Enviar DM
1	Melissa Alexandra	Lopez Ramirez	64229837	melissa_lopez@utec.com	2589232019	Es linda

Ilustración AA. Contactos

De igual forma, si el usuario presiona el botón de “Mis notas”, lo redirige al formulario de notas.

Notas

Hola, visualiza, guarda, edita y borra tus notas

Notas

Título:

Nota:

	Título	Notas
9	OCOM	Revisar las notas de los foros
3	Hacer lab	El laboratorio de física vence el esta semana
2	Pendiente Redes	Investigar redes
1	Progra	Terminar documentacion

Ilustración BB. Notas

Conclusiones

En conclusión, la herramienta facilita al campo estudiantil ayudando a organizar y administrar actividades presentes y futuras por medio de un bloc de notas; como agendar notas y agregar personas a la lista de contactos y actividades académicas asignadas por los docentes creando así un ambiente exclusivo

Dando las soluciones necesarias para la aplicación cumpliendo con los objetivos planteados en el documento. Manteniendo la simplicidad para aquellos que solo buscan crear notas de forma rápida.

Recomendaciones

Establecer un lugar específico para universidad y notaciones de clases ahora es posible, somos la alternativa, al utilizar aplicaciones de terceros en las que se puede distraer y/o discontinuarse de los que se está realizando.

- Utilizar nuestra aplicación, es segura y compatible con la plataforma Windows, donde puedes registrarte para tener únicamente acceso a todas tus notas o contactos. Tienes la facilidad de plasmar tu información estudiantil.
- En el apartado de contactos debes guardar los datos de tus amigos y/o compañeros de universidad con las que compartirás en la asignatura cursante.
- En el apartado de las notas, ya que en ellas se plasman todas las notaciones sobresalientes, apuntes, pendientes, importantes y notas como tal, con la posibilidad de integrar la fecha de dicha nota.
- Lo que se busca es: tener la mejor accesibilidad de parte de los estudiantes, para que se pueda llevar orden y documentación de la experiencia universitaria.

Tabla de Ilustraciones

Ilustración A. Tabla Alumno.....	8
Ilustración B. Procedimiento almacenado de Login.....	8
Ilustración C. Procedimientos almacenados para agregar	9
Ilustración D. Procedimiento almacenado para actualizar alumno	9
Ilustración E. Procedimiento almacenado para buscar alumno	10
Ilustración F. Procedimiento almacenado para eliminar alumno	10
Ilustración G. Tabla contactos	10
Ilustración H. Procedimiento almacenado para insertar contacto	11
Ilustración I. Procedimiento almacenado para actualizar contacto	11
Ilustración J. Procedimiento almacenado para eliminar contacto	12
Ilustración K. Tabla de notas	12
Ilustración L. Procedimiento almacenado para insertar nota	12
Ilustración M. Procedimiento almacenado para actualizar nota.....	13
Ilustración N. Procedimiento almacenado para eliminar una nota.....	13
Ilustración O. Procedimiento almacenado para consultar los contactos de alumno.....	13
Ilustración P. Procedimiento almacenado para consultar las notas de alumno	14
Ilustración Q. Procedimiento almacenado para obtener la cantidad de contactos de alumno.....	14
Ilustración R. Procedimiento almacenado para obtener la cantidad de notas de alumno.....	14
Ilustración S. Procedimiento almacenado para obtener la cantidad de carnet existentes.....	15
Ilustración T. Procedimiento almacenado para obtener la cantidad de emails existentes.....	15
Ilustración U. Procedimientos almacenados para procesar una nota de manera automática al estar registrado	15

Ilustración V. Diagrama de la base de datos generado por MySQL	16
Ilustración W. Proceso del archivo .jar	17
Ilustración X. JDialogForm	18
Ilustración Y. Agregar un JDialogForm	19
Ilustración Z. Home	20
Ilustración AA. Contactos	21
Ilustración BB. Notas	21