UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

BASE DE DATOS

PROYECTO FINAL

CONSTRUCCIÓN DE UNA BASE DE DATOS

INTEGRANTE

CRUZ VARGAS EMILIO
LUNA VELAZQUEZ SAID JOSUE
MARTINEZ PAVON MARIA GUADALUPE
RUIZ SANCHEZ MIGUEL ANGEL

PROFESOR

ING. FERNANDO ARREOLA FRANCO

GRUPO

1

Índice

1. INTRODUCCIÓN

2. PLAN TRABAJO

3. CRONOGRAMA

4. PLANEACION

5. MODELO RELACIONAL

6. FUNCIONALIDAD DE LA APP

7. CONCLUSIONES

Objetivos

El objetivo principal es desarrollar e implementar un sistema de gestión integral apoyado en una base de datos para el restaurante. Este sistema tiene como propósito mejorar la eficiencia operativa en todas las áreas del restaurante, desde la gestión de inventario, el control del personal, la administración de pedidos y reservas.

Propósito

Optimizar la gestión de información, mejorar la eficiencia operativa y proporcionar un servicio de calidad tanto a los clientes como al personal del establecimiento.

Alcance

El proyecto abarcará desde el diseño y la implementación de la base de datos hasta su despliegue y puesta en marcha en la aplicación del restaurante. Incluirá el desarrollo de interfaces de usuario para facilitar la interacción con la base de datos.

Las funciones de la aplicación son una forma de gestión de personal en donde se registra y administra información detallada del personal.

Atención al cliente, registra datos de los clientes para un servicio personalizado, además de controlar órdenes y facturación.

Gestión de productos, mantener un catálogo actualizado de platillos, bebidas y suministros, incluyendo detalles como precios, disponibilidad y categorías.

Control de inventarios, se encarga de supervisar y mantener niveles de existencias precisos.

Perspectivas del Proyecto

Operativa:

Eficiencia mejorada: Agilizar la toma de pedidos, la gestión de inventario y el seguimiento de la información del personal para optimizar las operaciones diarias del restaurante.

Reducción de errores: Minimizar errores en los pedidos, pagos y registros, mejorando la precisión y confiabilidad de los procesos operativos.

Experiencia del Cliente:

Servicio personalizado: Permitir la personalización de la experiencia del cliente al almacenar preferencias y datos históricos para una atención más individualizada.

Atención Ágil: Mejorar la rapidez y precisión en la toma de pedidos y la entrega de alimentos para una experiencia positiva del cliente.

Gestión Estratégica:

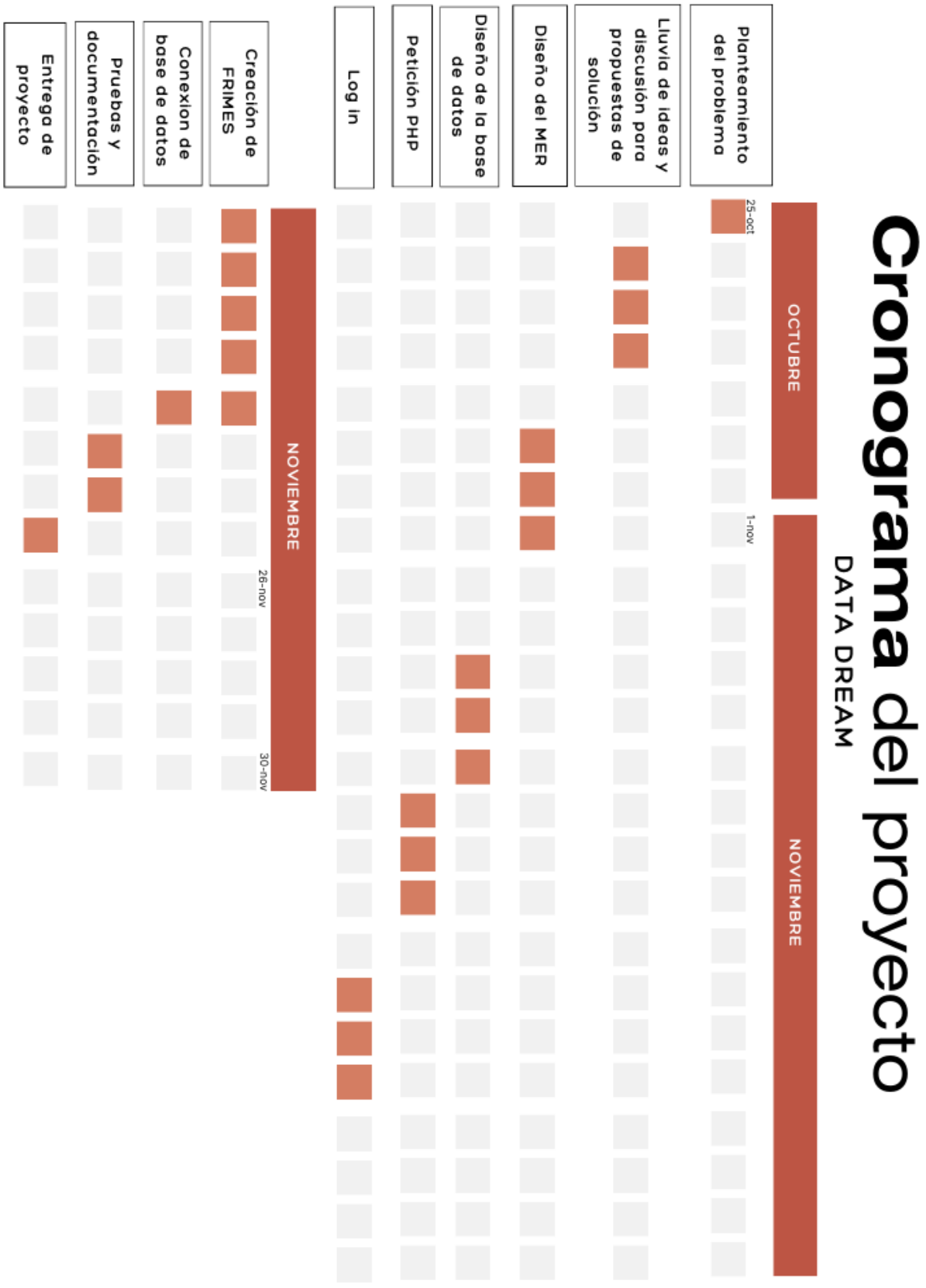
Toma de decisiones informada: Proporcionar informes y análisis detallados para tomar decisiones estratégicas basadas en datos reales sobre el desempeño del restaurante.

Optimización de recursos: Identificar áreas de mejora y oportunidades de crecimiento para optimizar recursos y aumentar la rentabilidad.

Funcionalidad

Gestión de Personal, en donde se tiene un registro y administración de datos del personal, asignación de roles, horarios laborales, salarios y detalles de contacto, lleva el registro de clientes en donde almacena información detallada sobre clientes habituales y ocasionales, lleva el control de los productos en existencia, mantenimiento de un catálogo completo de platillos, bebidas y otros productos ofrecidos por el restaurante. Detalles como nombre, descripción, precio, categoría y disponibilidad de cada ítem, permite tomar órdenes y facturar órdenes realizadas por los clientes, incluyendo detalles de los artículos seleccionados, precios y fechas de la orden.

CRONOGRAMA Y PERSONAL



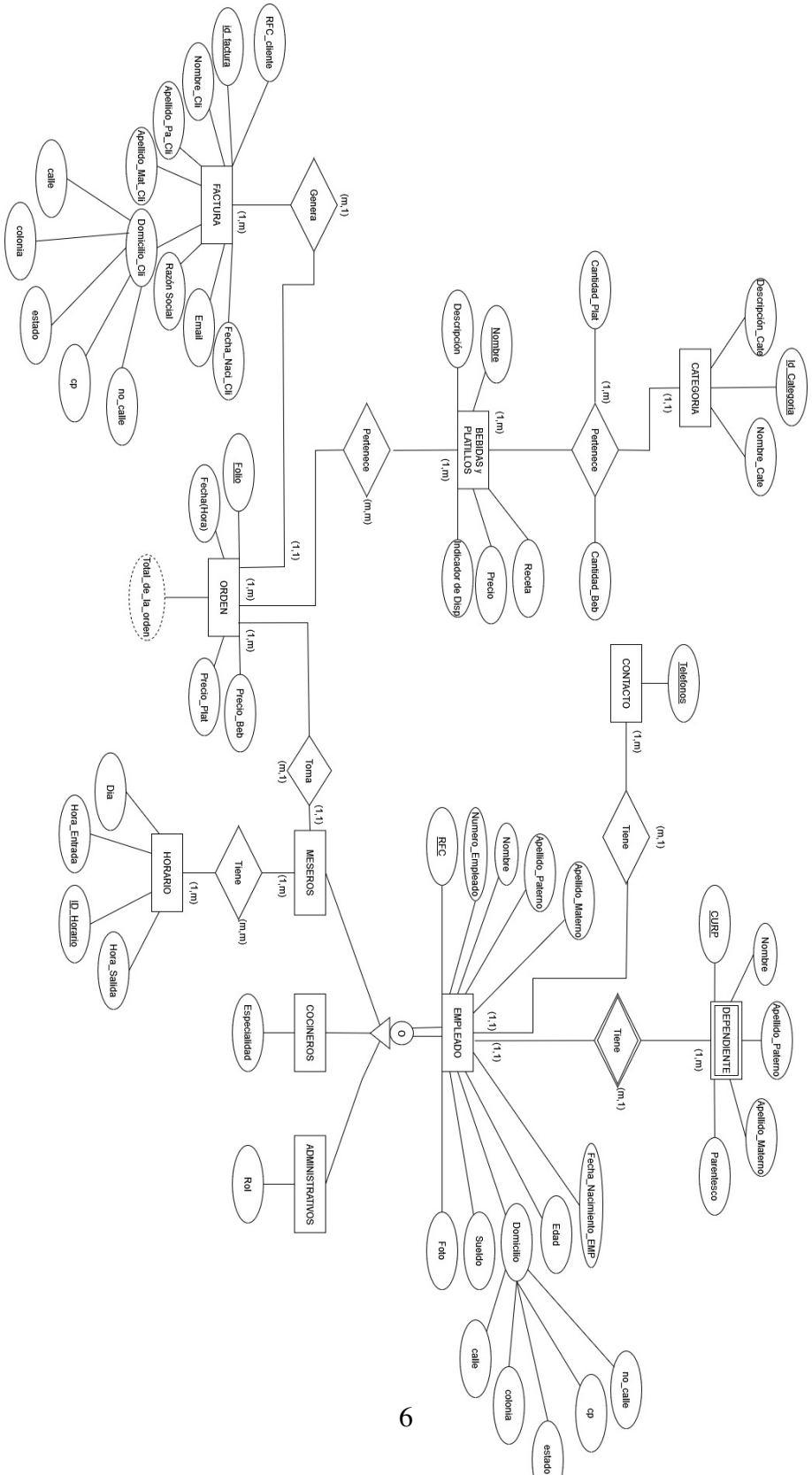
Nombre	Emilio Cruz Vargas
ROL	Desarrollador de software
Categoría profesional	Ingeniero en computación
Responsabilidades	Creación de la aplicación móvil e implementación de la página web
Información del contacto	emilio.cruz.vargas123@gmail.com
Aprobación	DATA DREAM

Nombre	Said Josue Luna Velazquez
ROL	Programador y diseñador de la base de datos
Categoría profesional	Ingeniero en computación
Responsabilidades	Diseño de los modelos MER y programación de la base de datos
Información del contacto	saidjosue137@gmail.com
Aprobación	DATA DREAM

Modelo relacional

Nombre	María Guadalupe Martínez Pavón
ROL	Quality Assurance
Categoría profesional	Ingeniera en computación
Responsabilidades	Documentar el proceso del proyecto, revisar errores y funcionalidad
Información del contacto	lupitampavon@gmail.com
Aprobación	DATA DREAM

Nombre	Miguel Ángel Ruiz Sanchez
ROL	Diseñador y Programación de base de datos
Categoría profesional	Ingeniero en computación
Responsabilidades	Programación complementaria para la aplicación móvil, manejo de la base de datos
Información del contacto	mike321755@hotmail.com
Aprobación	DATA DREAM



Se maneja un MER, el cual se va a dividir en una entidad llamada EMPLEADO, en donde tiene una relación de muchos a uno con la entidad CONTACTO, lo que significa que un empleado puede tener múltiples contactos, pero un contacto está asociado a un único empleado.
Esta misma entidad tiene una generalización en donde se incluyen las entidades de MESERO, COCINERO Y ADMINISTRATIVO, estas van a compartir rfc ya que se toma como llave primaria, número de empleado, nombre empleado, apellido paterno, apellido materno, fecha nacimiento, edad, calle, colonia, estado, código postal, número calle, sueldo y al mismo tiempo cada una de ellas tendrán su propios atributos; la entidad COCINERO tendrá el atributo "especialidad", ADMINISTRATIVOS tendrá "rol" y el MESERO tiene dos relaciones una con la entidad de HORARIO, esta incluye todos los datos de entrada, salida e información del turno que tomó el mesero y ORDEN cuenta con los atributos de folio, fecha hora, total de la orden (la cual va a poder ser calculable a partir de lo que los platillos y bebidas que se consumieron), precio platillos, precio bebidas, rfc.

La entidad ORDEN tiene dos relaciones más, una es con la entidad FACTURA que cuenta con los siguientes atributos id factura (PK), rfc cliente, nombre cliente, apellido paterno, apellido materno, calle, colonia, estado, código postal, número, calle, razón social, email, fecha nacimiento, folio orden y que al mismo tiempo tiene una relación con BEBIDAS Y PLATILLOS, que estos al mismo tiempo nombre, descripcion, receta, precio, indicador de disponibilidad, cantidad bebida, cantidad platillo, id categoria. y tiene una relación con CATEGORIA la cual tiene atributos como id categoria, nombre y descripción.

FRAME 1. MainActivity

Es una implementación de una actividad en Android, que gestiona una interfaz de inicio de sesión y realiza solicitudes HTTP´a un servidor remoto.

La interfaz de usuario se divide en dos campos de texto 'EditText' y 'editPassText', utilizados para ingresar el correo electronico y la contraseña.

```
EditText editEmpleadoText, editPassText;
```

```
editEmpleadoText = (EditText) findViewById(R.id.textEmailAddress);
editPassText = (EditText) findViewById(R.id.textPassword);
```

Un botón 'Button' llamado 'buttonMain' que se utiliza para iniciar sesión

```
Button buttonMain;
```

```
buttonMain = (Button) findViewById(R.id.textLogin);
buttonMain.setOnClickListener(new View.OnClickListener() {
```

Se aplico un método llamado 'onCreate()', que se ejecuta cuando una actividad se crea por primera vez, establece el diseño de la actividad utilizando 'setContentView(R.Layout.activitymain);'

Asigna los elementos de la interfaz gráfica (Los campos de texto y el botón) a las respectivas variables en el código mediante findViewById, define un OnClickListener para el botón buttonMain. Cuando se hace clic en este botón, se inicia un hilo (Thread tr) para realizar una solicitud al servidor.

```
oText = (EditText) findViewById(R.id.textEmailAddress);
t = (EditText) findViewById(R.id.textPassword);
= (Button) findViewById(R.id.textLogin);
setOnClickListener(new View.OnClickListener() {
```

Función enviarPost(String cor, String pas), Construye una cadena de parámetros (parametros) con el correo (cor) y la contraseña (pas) y establece una conexión HTTPS con una URL específica y realiza una solicitud POST al servidor con los parámetros.

```
public String enviarPost(String cor, String pas){
    String parametros="cor="+cor+"&pas="+pas;
    HttpsURLConnection conection=null;
    String respuesta="";
    try{
        URL url=new URL( spec: "https://bdf1.campanita.xyz/index.php");
        conection = (HttpsURLConnection) url.openConnection();
        conection.setRequestMethod("POST");
        conection.setRequestProperty("Content-Length",""+Integer.toString(parametros.getBytes().length));

        conection.setDoOutput(true);
        DataOutputStream wr=new DataOutputStream(conection.getOutputStream());
        wr.writeBytes(parametros);
        wr.close();

        Scanner inStream=new Scanner(conection.getInputStream());
        while(inStream.hasNextLine())
            respuesta+=(inStream.nextLine());
    }catch(Exception e){}
    Log.d( tag: "hola",respuesta.toString());
    return respuesta.toString();
}
```

Función objJSON(String rspta que toma la respuesta del servidor en formato de cadena (respuesta) y la intenta convertir en un JSONArray, si el JSONArray tiene una longitud mayor que cero, devuelve 1, indicando una respuesta positiva; de lo contrario, devuelve 0.

```
public int objJSON(String rspta){
    int res=0;
    try{
        JSONArray json=new JSONArray(rspta);
        if(json.length()>0)
            res=1;
    }catch(Exception e){}
    return res;
}
```

El código usa HttpsURLConnection para realizar la solicitud HTTP. Esta operación se ejecuta en el hilo principal, lo cual no es recomendado ya que podría bloquear la interfaz de usuario si la solicitud tarda mucho tiempo. En la práctica, se debería realizar en un hilo separado (no en el hilo principal) para no afectar la capacidad de respuesta de la aplicación.
Las excepciones (catch(Exception e)) se manejan de manera genérica, lo cual no es una buena práctica. Es recomendable manejar los tipos de excepciones específicas para realizar una gestión adecuada de errores.

```
    Scanner inStream=new Scanner(conection.getInputStream());
    while(inStream.hasNextLine())
        respuesta+=(inStream.nextLine());
}catch(Exception e){}
Log.d( tag: "hola",respuesta.toString());
return respuesta.toString();
```

El código utiliza el Log.d para imprimir la respuesta del servidor en el log. Esto es útil para depuración, pero en una aplicación real, podría ser preferible utilizar registros más estructurados o enviar errores a un sistema de registro específico.

```
Log.d( tag: "hola",respuesta.toString());
return respuesta.toString();
```

FRAME 2. Principal
Este código tiene como objetivo recopilar datos del usuario a través de una interfaz de usuario, validar la información ingresada y enviarla a un servidor remoto para su procesamiento o almacenamiento.
La clase Principal contiene una serie de variables (String) que almacenan la información ingresada por el usuario en diferentes campos como nombre, apellidos, edad, dirección, RFC, salario, etc.

```
public class Principal extends AppCompatActivity {
    5 usages
    String nombreIngresado;
    5 usages
    String apMaIngresado;
    5 usages
    String apPaIngresado;
    5 usages
    String edadIngresada;
    5 usages
    String fechaNacimientoIngresada;
```

Método onCreate(),este método se ejecuta al inicio de la actividad y se encarga de inicializar la interfaz gráfica con los elementos definidos en el layout oficialsecondframe.xml. , obtiene referencias a los campos de texto (EditText) y al botón de guardado, establece un OnClickListener para el botón de guardado (saveButton). Cuando se hace clic en este botón, se recopila la información ingresada por el usuario de los campos de texto.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.oficialsecondframe);

    // Obtener referencias a los EditText
    editTextNombre = findViewById(R.id.setName);
    editTextApMa = findViewById(R.id.setApMa);
    editTextApPa = findViewById(R.id.setApPa);
    editTextEdad = findViewById(R.id.setEdad);
    editTextFechaNacimiento = findViewById(R.id.setFeNa);
    editTextRFC = findViewById(R.id.setRFC);
    editTextDireccionEstado = findViewById(R.id.setDireccionEstado);
    editTextDireccionColonia = findViewById(R.id.setDireccionColonia);
    editTextDireccionCalle = findViewById(R.id.setDireccionCalle);
    editTextDireccionNumCa = findViewById(R.id.setDireccionNumCa);
    editTextDireccionCP = findViewById(R.id.setDireccionCP);
    editTextSalario = findViewById(R.id.setSalario);
```

Funciones de validación, validarFormatoNombre, validarFormatoApellido, validarEdad, validarFechaNacimiento, validarRFC, validarDireccion, validarSalario: Estas funciones realizan diferentes validaciones sobre los datos ingresados por el usuario. Verifican que los datos cumplan con ciertos criterios (como formato, longitud, si son valores numéricos enteros, etc.).

```
// Validar el formato de nombre, apellidos, edad, fecha de nacimiento, RFC, dirección y salario
if (validarFormatoNombre(nombreIngresado) &&
        validarFormatoApellido(apMaIngresado, apPaIngresado) &&
        validarEdad(edadIngresada) &&
        validarFechaNacimiento(fechaNacimientoIngresada) &&
        validarRFC(rfcIngresado) &&
        validarDireccion(direccionEstadoIngresada, direccionColoniaIngresada, direccionCalleIngresada,
        validarSalario(salarioIngresado)) {
```

Envío de datos al servidor, dentro del método onClick del botón de guardado, se recopilan los datos ingresados por el usuario y se realizan validaciones utilizando las funciones mencionadas anteriormente, si los datos son válidos, intenta realizar una solicitud a un servidor remoto utilizando HttpURLConnection. Se establece una conexión con una URL específica y se realiza una solicitud POST con los parámetros correspondientes (datos del usuario), se manejan los posibles códigos de respuesta del servidor (HTTPOK, HTTPINTERNALERROR y otros), además, se utiliza Volley (una biblioteca de Android para realizar solicitudes HTTP de manera más sencilla) para enviar los datos a través de una solicitud POST al mismo servidor. Se crea un StringRequest con los parámetros a enviar y se agrega a una RequestQueue para su ejecución.

```
// Validar el formato de nombre, apellidos, edad, fecha de nacimiento, RFC, dirección y salario
if (validarFormatoNombre(nombreIngresado) &&
        validarFormatoApellido(apMaIngresado, apPaIngresado) &&
        validarEdad(edadIngresada) &&
        validarFechaNacimiento(fechaNacimientoIngresada) &&
        validarRFC(rfcIngresado) &&
        validarDireccion(direccionEstadoIngresada, direccionColoniaIngresada, direccionCalleIngresada,
        validarSalario(salarioIngresado)) {
```

Función mostrarMensajeError(), muestra mensajes de error mediante Toast cuando se detecta un problema en la validación de los datos. Funciones auxiliares, esEntero(): Verifica si una cadena puede convertirse en un número entero.
esFormatoFechaValido(): Verifica si una cadena tiene un formato de fecha válido.

```
    private boolean validarFechaNacimiento(String fechaNacimiento) {
        // Verificar que la fecha de nacimiento no sea nula, no esté vacía y tenga el formato adecuado
        return fechaNacimiento != null && !fechaNacimiento.isEmpty() && esFormatoFechaValido(fechaNacimiento);
    }

    1 usage
    private boolean esFormatoFechaValido(String fecha) {
        // Definir el formato de fecha esperado (en este caso, el formato utilizado en PostgreSQL)
        SimpleDateFormat formatoFecha = new SimpleDateFormat( pattern: "yyyy-MM-dd", Locale.getDefault());
        formatoFecha.setLenient(false);
```

Conclusiones

Cruz Vargas Emilio

La conclusión obtenida de este proyecto revela la universalidad de la implementación de bases de datos. La necesidad de almacenar información es constante y varía según la perspectiva que se le otorgue, dándole el valor y significado que buscamos. Es crucial mantener la integridad y seguridad de estos datos.

Este proyecto, en particular, me brindó una comprensión más clara sobre el comportamiento de una base de datos. Ahora entiendo cómo acceder a ella, interpretar la información y utilizarla en aplicaciones. Además, destaco la importancia del diseño y la correcta implementación de una base de datos, ya que simplifica su gestión. Concentrarse en la interconexión entre áreas en lugar de resolver errores específicos resulta fundamental, ha sido esclarecedor en cuanto a las aplicaciones prácticas de una base de datos, ofreciendo un ejemplo realista que puede encontrarse en situaciones cotidianas

Luna Velazquez Said Josue Durante la realización de este proyecto, tuve la oportunidad de aplicar activamente muchos de los conceptos vistos en clase, lo que resultó en la resolución exitosa de diversas tareas. Sin embargo, me encontré con desafíos notables, como la indecisión en la elaboración de los modelos. En varias etapas, al revisar las versiones creadas, surgió la posibilidad de mejoras que podrían facilitar la implementación. A pesar de estas indecisiones iniciales, el proceso de revisión constante me enseñó la importancia de la flexibilidad y la mejora continua en el desarrollo de modelos. Además, la adaptación a un nuevo entorno de desarrollo fue un desafío significativo.La curva de aprendizaje fue abrupta, pero la rápida familiarización con el entorno fue esencial para seguir contribuyendo al proyecto. A pesar de las dificultades, esta transición resultó ser una elección acertada, ya que permitió lograr un resultado visual impresionante en la implementación final.

Estos desafíos no solo representaron obstáculos, sino valiosas oportunidades de aprendizaje. Me enseñaron la importancia de la flexibilidad, la adaptabilidad y la búsqueda constante de mejoras, elementos fundamentales para futuros proyectos.

Martínez Pavón María Guadalupe

Durante el desarrollo de esta práctica fue un poco complicado ver como se utilizaban las diferentes clases dentro de Android Studio, pues nunca había ocupado la interfaz ni sabia como programar, pero gracias al trabajo en equipo y al apoyo de los compañeros fue más facil visualizarlo, otra cosa que nos dimos cuenta fue que algunas aplicaciones de programación si bien son un tanto golbales, el hecho de tener que trabajar en equipo pero cada uno en forma remota si resultó un reto, ya que el dividiernos el trabjo fue buena idea, hasta que vimos que se tenian que implementar en la aplicación, este proyecto me sirvió para dar una repasada a los temas vistos en clase, ya que se retomaron muchas cosas.

Agradezco a mis clases de laboratorio que tambien me dió una visión clara para la solución de problemas dentro de la base de datos, ya que el manejo de una base de datos, es un tanto complicada pero a la hora de retomar los conceptos fue más sencillo ver el esquema de la base de datos

Ruiz Sanchez Miguel Angel

Para este proyecto mi enfoque se centro en la creacion e implementación del diseño de la base de datos, de la cual busque la forma de complementar el diseño aportando ideas a través de la generación de un nuevo modelo entidad relacion extendido (MERE) gracias a esto pudimos obtener el modelo final, el cual nos fue util para la siguiente etapa de implementación del modelo logico.

Finalmente al implementar el modelo logico aplicando conocimientos de DDL y DML, habiamos llegado a tener una buena visión en la integridad de los datos permitiendonos asi tener una mejor gestion en la base de datos.