



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA — ING. EN COMPUTACIÓN

BASES DE DATOS 1644
GRUPO 01 — 2021-1

PROYECTO FINAL: DISEÑO E IMPLEMENTACIÓN DE BASE DE DATOS

Realizado por: Zamora Ramírez Oswaldo Rafael, Sánchez
Manjarrez Andrew y Ramírez Vigueras Alma Yeremay

Supervisado por:
Ing. Fernando Arreola Franco

Índice

1. REQUISITOS DE PROYECTO	2
1.1. Objetivo	2
1.2. Descripción del Problema	2
1.3. Entregables	3
2. INTRODUCCIÓN	5
3. PLAN DE TRABAJO	6
3.1. Proposito y Requerimientos de Alto Nivel	6
3.2. Cronograma de Hitos	6
3.3. Descripción de Alcances	7
3.4. Designación de Roles y Responsabilidades	7
4. DESARROLLO	10
4.1. Diseño Preliminar	10
4.1.1. Modelo Entidad Relación	10
4.2. Fase de Aplicación-Web	12
4.2.1. BackEnd	12
4.2.2. FrontEnd	13
5. IMPLEMENTACIÓN	14
5.1. Diagrama de Conexiones	14
5.1.1. Descripción de las funciones	15
6. CONCLUSIONES	18
7. PRESENTACIÓN	20
7.1. Ejecución de API	20
7.2. Prueba de Aplicación	20

1. REQUISITOS DE PROYECTO

1.1. Objetivo

El alumno analizará una serie de requerimientos y propondrá una solución que atienda a los mismos, aplicando los conceptos vistos en el curso.

1.2. Descripción del Problema

PARTE UNO:

Consiste en el diseño de una base de datos. Una cadena de papelerías busca innovar la manera en que almacena su información, y los contratan para que desarrollen los sistemas informáticos para satisfacer los siguientes requerimientos: Se desea tener almacenados datos como la razón social, domicilio, nombre... y teléfonos de los proveedores, razón social, nombre, domicilio y al menos un email de los clientes. Es necesario tener un inventario de los productos que se venden, en el que debe guardarse el código de barras, precio al que fue comprado el producto, fecha de compra y cantidad de ejemplares en la bodega (stock). Se desea guardar la marca, descripción y precio de los regalos, artículos de papelería, impresiones y recargas, siempre y cuando se tenga su correspondiente registro en el inventario. Debe también guardarse el número de venta, fecha de venta y la cantidad total a pagar de la venta, así como la cantidad de cada artículo y precio total a pagar por artículo. Además, se requiere que:

- Al recibir el código de barras de un producto, regrese la utilidad.
- Cada que haya la venta de un artículo, deberá decrementarse el stock por la cantidad vendida de ese artículo. Si el valor llega a cero, abortar la transacción. Si hay menos de 3, emitir un mensaje.
- Dada una fecha, o una fecha de inicio y fecha de fin, regresar la cantidad total que se vendió en esa fecha/periodo
- Permitir obtener el nombre de aquellos productos de los cuales hay menos de 3 en stock.
- De manera automática se genere una vista que contenga información necesaria para asemejarse a una factura de una compra.
- Crear al menos, un índice, del tipo que se prefiera y donde se prefiera. Justificar el porqué de la elección en ambos aspectos.

Tomar en cuenta las siguientes consideraciones:

- Puede haber distintas soluciones al problema.

- Los requerimientos enlistados anteriormente, deberán ser realizados por medio de PgSQL, con los elementos que se consideren adecuados para resolverlos.
- El número de venta debe tener un formato similar a "VENT-001", prefijo VENT, seguido de un guión y un número secuencial
- Donde este presente el atributo domicilio, esta compuesto por estado, código postal, colonia, calle y numero
- El diseño debe satisfacer todos los principios de diseño, los requerimientos anteriores y un buen manejo de información.

PARTE DOS:

Una vez diseñada y lista la base de datos, se debe crear una interfaz gráfica vía app móvil o web, que permita:

- Agregar la información de un cliente.
- Ingresar una venta, de hasta 3 artículos, los cuales podrán seleccionarse de una lista de opciones, permitir ingresar la cantidad, calcular el costo total de cada artículo y el costo total de toda la venta. Ingresar dicha información en la base de datos, respetando todas las restricciones de integridad.

1.3. Entregables

Es necesario entregar lo que se solicita a continuación:

DOCUMENTO: Documento Se deberá entregar un documento FORMAL elaborado en Latex, que contenga las siguientes secciones:

- Introducción: Breve descripción del análisis del problema, su(s) objetivos, su propuesta(s) de solución.
- Plan de trabajo: Queda a criterio de cada equipo el grado de detalle que se le dará a esta sección, pero mínimo debe contener descripción general de las actividades a realizar y su correspondiente plan de actividades, así como un breve detalle de qué hizo cada miembro del equipo.
- Diseño: Descripción de lo realizado en las correspondientes fases de diseño de las bases de datos, agregando los resultados de cada una de ellas.
- Implementación: Descripción de funcionamiento y código sólo de los stored procedures, triggers, funciones, etc. Empleados para cumplir con los requerimientos del problema.
- Presentación: Descripción de lo que hace la modalidad seleccionada como forma de conexión hacia la base de datos

- Conclusiones: Personales, detallando las dificultades, retos, aciertos, etc. que se presentaron en el proyecto.

CODIGO: A través de github, se deberá crear una carpeta por equipo donde se anexe lo siguiente, asignando nombres ADECUADOS para cada documento/archivo:

- El documento del punto anterior
- Códigos fuente de MER, MR
- Script de creación de la base de datos y tablas
- Script para el agregado de información
- Script de toda la programación a nivel BD
- Códigos de lo implementado como parte de la etapa de presentación
- La presentación a emplear a la hora de exponer

EXPOSICIÓN

Como parte de la evaluación del curso, se deberá realizar una presentación del proyecto. La idea es simular la presentación de una solución de software a un posible cliente, lo que implica descripción del software, muestras de funcionalidad, solución de dudas, etc. Cada equipo es libre de decidir cómo hacer su presentación, el único detalle es que debe ser TOTALMENTE formal.

2. INTRODUCCIÓN

Las bases de datos son herramientas muy poderosas que se utilizan en todas las áreas de la computación. Es una habilidad informática clave poder organizar datos, crear bases de datos y controlar datos utilizando lenguajes de consulta. Uno de los principales beneficios de las bases de datos informáticas es que facilitan el almacenamiento de información para que sea rápido y fácil de encontrar.

El uso que se le pueda dar a una base para convertir los datos en información útil es amplia y variada. En este caso la cadena de Papelerías UTP requiere modernizar su sistema de almacenamiento de datos para una estructura de ventas. El problema puede satisfacerse con un modelo escalable abstraído en una aplicación móvil.

Los requerimientos mínimos que desean tener almacenados datos como la razón social, domicilio, nombre y teléfonos de los proveedores, razón social, domicilio y al menos un email de los clientes. Es necesario tener un inventario de los productos que se venden, en el que debe guardarse el código de barras, precio al que fue comprado el producto, fecha de compra y cantidad de ejemplares en la bodega (stock). Se desea guardar la marca, descripción y precio de los regalos, artículos de papelería, impresiones y recargas, siempre y cuando se tenga su correspondiente registro en el inventario. Debe también guardarse el número de venta, fecha de venta y la cantidad total a pagar de la venta, así como la cantidad de cada artículo y precio total a pagar por artículo.

La resolución requiere además del uso de procedures, funciones y triggers; Los procedimientos no devuelven ningún valor, solo obtienen parámetros y hacen algo con ellos, las funciones hacen lo mismo porque también pueden devolver un valor basado en la información. Los triggers son una especie de controladores de eventos que reaccionan a cualquier acción que se desee e inician el procedimiento cuando ocurre esta acción.

3. PLAN DE TRABAJO

3.1. Proposito y Requerimientos de Alto Nivel



El siguiente plan de trabajo presenta una compilación de apartados que demuestran una oportunidad para ampliar las aptitudes. Ante la necesidad de los interesados de recurrir a la sofisticación de técnicas generalizadas para el diseño, manejo y mantenimiento de las Bases de Datos se describen las siguientes actividades para evaluar las habilidades adquiridas durante el curso.

3.2. Cronograma de Hitos

ACTIVIDAD	DURACIÓN	S1	S2	S3	S4	S5	S6	S7
FASE: PLANEACIÓN	7 hrs							
Análisis Requisitos	1 hrs.							
Gestión Alcances y Riesgos	1 hrs.							
Planeación y Diseño Modular	2 hrs.							
Designación de Roles y Responsabilidades	1 hr.							
Selección de Herramientas	2 hr.							
FASE: DESARROLLO	52 hrs.							
Diseño MER	2 hrs.							
Documentación Plan de Trabajo	10 hrs.							
Metadología usada	2 hrs.							
Codificación	30 hrs.							
Búsqueda de Campo	2 hrs.							
Filtrado y Análisis	6 hrs.							
FASE: IMPLEMENTACIÓN	32 hrs.							
Pruebas Preliminares	4 hrs.							
Edición y Debugging	10 hrs.							
Creación de la Expo	4 hrs.							
Documentación	4 hrs.							
Pruebas Finales	6 hrs.							
Presentación	2 hrs.							
Lecciones y Conclusiones	2 hrs.							

Figura 1: El plan de trabajo establecido tendrá un periodo de 7 semanas funcionales y se dividira en 3 fases: Planeación, Desarrollo e Implementación, cada fase abarcara los puntos establecidos en los requisitos de proyecto.

3.3. Descripción de Alcances

Se responde a la necesidad de la cadena de papelerías UTP para dar solución a la renovación de su sistema de almacenamiento de datos a una respuesta moderna que incluya interfaces innovadoras.

3.4. Designación de Roles y Responsabilidades

Se empleó el sistema kanban, para el registro de actividades con tarjetas virtuales organizar tareas, agregar listas, y adjuntar archivos, así como etiquetar eventos, agregar comentarios y compartir tableros. Para esto utilizamos la herramienta virtual Trello que es un software de administración de proyectos con interfaz web y con cliente para iOS y android.

A continuación se muestra el tablero principal con acciones numeradas de colaboración de proyecto.

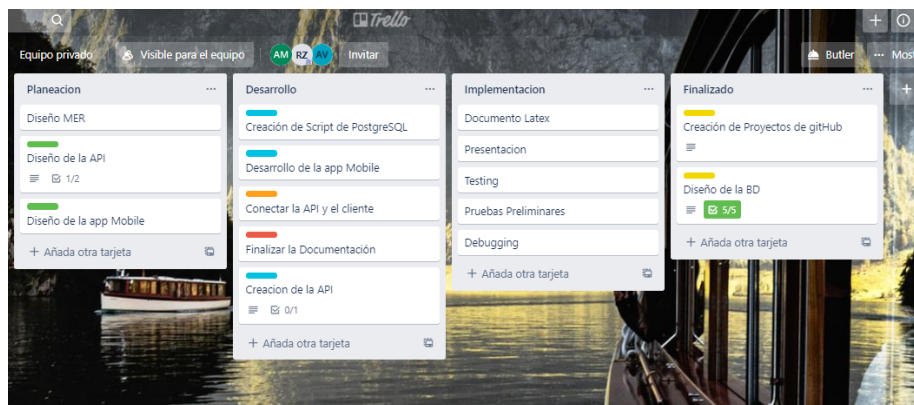


Figura 2: Los tableros con las fases destinadas contienen tareas asignadas con base en un código de colores que ayudan a indicar la asignación o supervisión de los integrantes. Con ayuda del cronograma de actividades se determinaron los hitos correspondientes de cada fase.

Se utilizaron diversos canales de comunicación formal e informal para compartir la información, agendar reuniones y lograr una mejor independencia descentralizada de trabajo.

Se acordaron y calendarizaron fechas de revisión y avance además de documentarse las observaciones obtenidas, así como la gestión de cambios.

Calendarización de Reuniones		
Plataforma	Fecha	Horario
Zoom	20/12/2020	18:00
Zoom	7/01/2021	9:00
Zoom	23/01/2021	18:00

Diseño de la BD

en la lista [Finalizado](#)

ETIQUETAS

Fase 1
Fase 0
+

Description
Editar

Se usara [diagrams.net](#) para el diseño de la BD el link del archivo es
<https://drive.google.com/file/d/1XX73QH8mWa1uimRQJhxsmyYm92Zmup46/view?usp=sharing>

Tareas
Ocultar elementos completados
Eliminar

100%

✓

Diseño del MER

✓

Revisar el MER

✓

Diseño del MR

✓

Revisar el MR

✓

Normalizar el MR hasta la 3ra forma normal

Añada un elemento

Activity
Mostrar detalles

AM

Escriba un comentario...

SUGERENCIAS

Unirse

AÑADIR A LA TARJETA

Miembros

Etiquetas

Checklist

Vencimiento

Adjunto

Portada

POWER-UPS

+ Añadir Power-Ups

BUTLER
NUEVO

+ Añadir botón en t...

ACCIONES

→ Mover

Copiar

Convertir en plant...

Figura 3: Ejemplo desglosado de paquete de trabajo 'Diseño de la BD', puede apreciarse la etiqueta de Fase, descripción con los enlaces de trabajo, comentarios, fechas y asignación de rol.

Se utilizó un sistema de control de versiones moderno con algún mecanismo para soportar el uso de ramas. En este caso GitHub. A diferencia de otros sistemas de control de versiones, Git promueve un ciclo de desarrollo donde las ramas se crean y se unen ramas entre sí, incluso varias veces en el mismo día permitiendo a los miembros del equipo una incorporación paralela de trabajo.

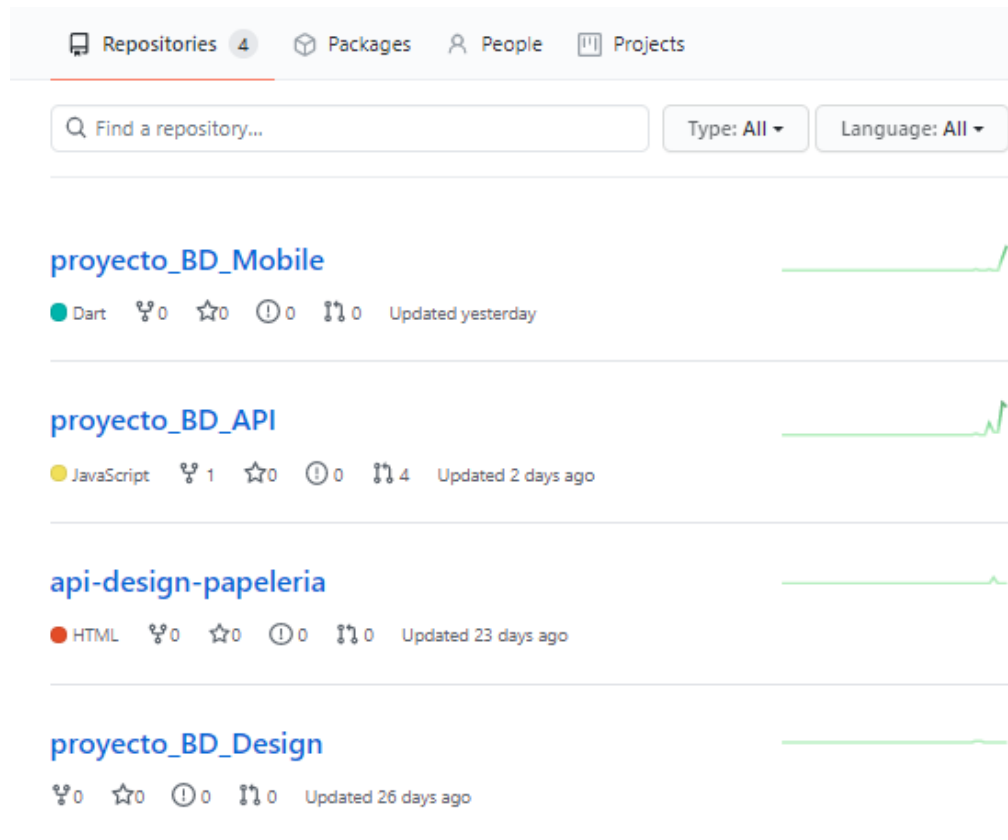


Figura 4: Los repositorios utilizados se nombraron acorde a la funcionalidad y se destaca el uso de lenguaje JavaScript y Dart, además de la utilización de entornos de desarrollo.

4. DESARROLLO

4.1. Diseño Preliminar

4.1.1. Modelo Entidad Relación

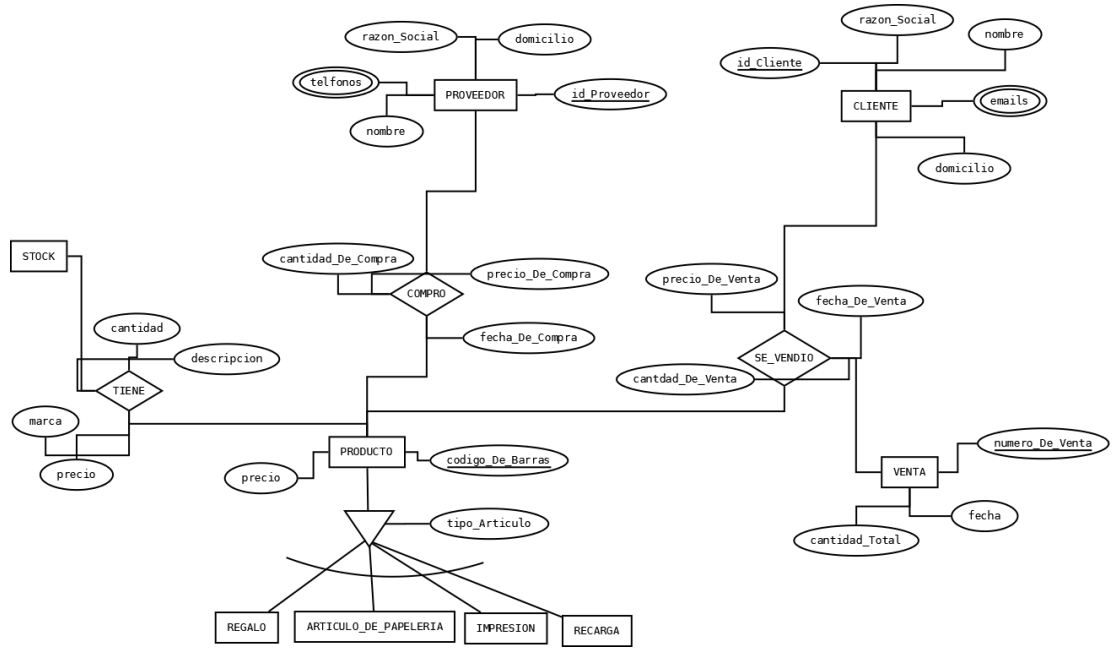


Figura 5: Diagrama Entidad relación extendido que denota el comportamiento de las entidades y su interacción para la red de Papelerías UTP.

Modelo MER Intermedio

- **PROVEEDOR:** $ID_{PROVEEDOR}$ smallintPK, $razon_social$ varchar(12), $nombre$ varchar(40), ap_pat varchar(40), ap_mat varchar(40), $estado$ varchar(40), cps smallint, $colonia$ varchar(40), $calle$ varchar(40), $numero$ smallint
- **TELEFONO:** $ID_{PROVEEDOR}$ smallintPK, $num_telefono$ varchar(10)
- **CLIENTE:** $ID_{CLIENTE}$ smallintPK, $razon_social$ int, $nombre$ varchar(40), $calle$ varchar(40), $numero$ smallint, $colonia$ varchar(40), $estado$ varchar(40), cps maillint
- **CORREO :** $ID_{CLIENTE}$ smallintPK, $email$ varchar(50)
- **COMPRA:** $ID_{PROVEEDOR}$ smallintPK, $cant_compra$ smallint, $precio_compra$ smallint, $fecha_compra$ date

- VENTA: $\text{num}_{\text{Ventas}} \text{smallint PK}, \text{precio}_{\text{Total}} \text{int}$
 $, \text{cantidad}_{\text{artTotal}} \text{smallint}, \text{precio}_{\text{artTotal}} \text{smallint}, \text{fecha} \text{date}$
- PRODUCTO: $\text{cod}_{\text{Barras}} \text{varchar}(13), \text{precio} \text{int}, \text{tipo}_{\text{Articulo}} \text{varchar}(13)$
- TIENE: $\text{cod}_{\text{Barras}} \text{varchar}(13) \text{PK}, \text{cantidad} \text{smallint}, \text{descripcion} \text{varchar}(100),$
 $\text{marca} \text{varchar}(40), \text{precio} \text{int}$
- COMPRA: $\text{ID}_{\text{PROVEEDOR}} \text{smallint PK}, \text{cant}_{\text{Compra}} \text{smallint},$
 $\text{precio}_{\text{Compra}} \text{smallint}, \text{fecha}_{\text{Compra}} \text{date}$


4.2. Fase de Aplicación-Web

Se dividió el diseño del proyecto en Frontend que es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Incluyendo todas las tecnologías de diseño y desarrollo web que corren en la aplicación y que se encargan de la interactividad con los usuarios. Y en Backend que es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El Backend también accede al servidor, que es una aplicación especializada que entiende la forma como el navegador solicita cosas.

4.2.1. BackEnd

Cuando hablamos de Backend nos referimos al interior de las aplicaciones que viven en el servidor y al que a menudo se le denomina “el lado del servidor”.


El backend de nuestra aplicación web consiste en un servidor remoto gratuito proveído por Heroku, una API y una base de datos. Se toman los datos, se procesa la información y se envía al usuario



reitmas32 Merge pull request #17 from Orshware/rafa_zam

2c4e6ff

14 hours ago



56 commits

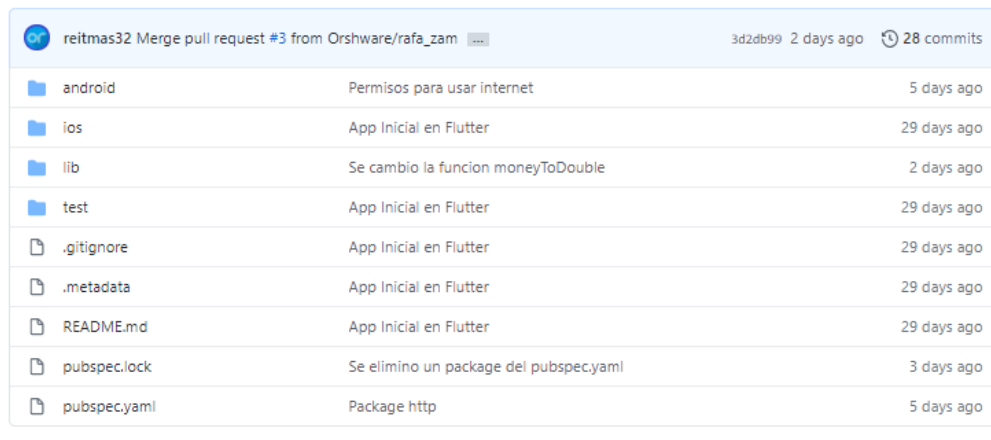
src	Se creo la connexcion a una base de datos remota	3 days ago
.babelrc	Uso de Babel	last month
.gitignore	gitignore	last month
PAPELERIA_CONFIG.sql	Script para crear usuario y data base	9 days ago
PAPELERIA_CREATE_DATABASE.sql	Se añadieron nuevas funciones a la base de datos y se refactorizo el ...	5 days ago
PAPELERIA_DELETE.sql	Script para eliminar la base de datos	9 days ago
PAPELERIA_FUNCS.sql	Se añadió la creacion de un indice	14 hours ago
PAPELERIA_INIT.sql	Se añadieron nuevas funciones a la base de datos y se refactorizo el ...	5 days ago
PAPELERIA_INSERT_DATA.sql	Se añadieron nuevas funciones a la base de datos y se refactorizo el ...	5 days ago
README.md	Instrucciones para compilar y correr	last month
database-script.SQL	Se comenzo a a crear el Script de la Base de Datos	23 days ago
package-lock.json	se añadió la biblioteca pg para la connexcion con postgresql	9 days ago
package.json	se añadió la biblioteca pg para la connexcion con postgresql	9 days ago
script.sh	se añadieron verbos a las rutas	7 days ago

Figura 6: El repositorio mostrado contiene las funciones de la Aplicación, si se desea trasladar la aplicacion a un sitio web, bastará solo con mover el esqueleto a una nueva anatomía Frontend.

4.2.2. FrontEnd

Front End es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que veremos en la pantalla cuando accedemos a la aplicación: tipos de letra, colores, adaptación para distintas pantallas(RWD), los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales. . . y otros elementos que permiten navegar dentro de una aplicación. Este conjunto crea la experiencia del usuario.

El desarrollador frontend se encarga de la experiencia del usuario, es decir, en el momento en el que este entra a una aplicación, debe ser capaz de navegar por ella, por lo que el usuario verá una interface sencilla de usar, atractiva y funcional.



The screenshot shows a GitHub pull request interface. At the top, it says "reitmas32 Merge pull request #3 from Orshware/rafa_zam" with a commit hash "3d2db99" and "2 days ago" and "28 commits". Below this is a table of files and their commit history.

android	Permisos para usar internet	5 days ago
ios	App Inicial en Flutter	29 days ago
lib	Se cambio la funcion moneyToDouble	2 days ago
test	App Inicial en Flutter	29 days ago
.gitignore	App Inicial en Flutter	29 days ago
.metadata	App Inicial en Flutter	29 days ago
README.md	App Inicial en Flutter	29 days ago
pubspec.lock	Se elimino un package del pubspec.yaml	3 days ago
pubspec.yaml	Package http	5 days ago

Figura 7: El repositorio mostrado contiene los parametros de diseño por los que navegará el usuario en su plataforma móvil.

5. IMPLEMENTACIÓN

5.1. Diagrama de Conexiones

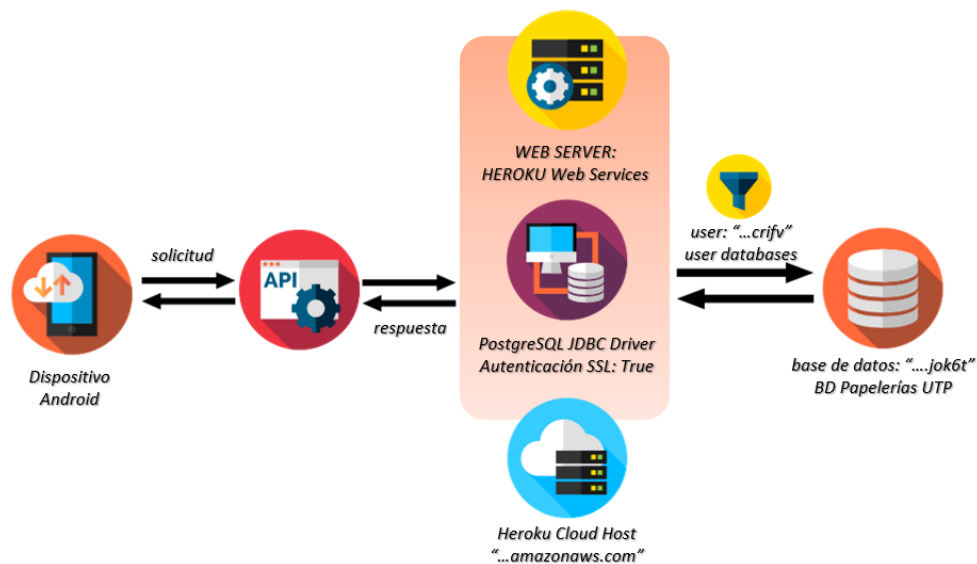


Figura 8: El diagrama de conexiones muestra el camino del proceso de solicitud hasta la respuesta que se obtuvo de la base de datos .

```
src > database.js > pool
1 import { Pool } from 'pg';
2
3 const pool = new Pool({
4   host: 'ec2-54-156-85-145.compute-1.amazonaws.com',
5   user: 'fmvxyqlwzcrifv',
6   database: 'd6gndvk95jok6t',
7   port: '5432',
8   password: '75cbbb7f7746b7af735919f4def1d9467455f3356f1c5807511ac25d3475e98',
9   ssl: { rejectUnauthorized: false }
10 });
11
12 export default pool;
```

Figura 9: La figura muestra las credenciales de acceso para los parametros de autenticación que se deben cumplir para la respuesta de la base de datos.

5.1.1. Descripción de las funciones

Descripción de funcionamiento y código sólo de los stored procedures, triggers, funciones, etc. Empleados para cumplir con los requerimientos del problema.

```
--1
--Al recibir el código de barras de un producto, regrese la utilidad.
--
CREATE FUNCTION utilidad(varchar(20)) RETURNS TABLE(utilidad money) AS
$$
SELECT precio_venta - precio_compra as utilidad
FROM producto
JOIN provee
ON producto.codigo_barras = provee.codigo_barras
WHERE producto.codigo_barras = $1
$$
LANGUAGE SQL;
```

Figura 10: Se muestra la función para el primer requerimiento, al recibir un código de barras, se debe regresar la utilidad.

```
CREATE FUNCTION validar_compra() RETURNS TRIGGER AS $validar_compra$
DECLARE
precio_art money ARRAY = ARRAY(SELECT precio_venta FROM producto WHERE producto.codigo_barras = NEW.codigo_barras LIMIT 1);
BEGIN
IF EXISTS(SELECT * FROM inventario_producto WHERE inventario_producto.codigo_barras = NEW.codigo_barras AND inventario_producto.cantidad = 0) THEN
RAISE EXCEPTION 'NO HAY EN EXISTENCIA %', NEW.codigo_barras;
ELSEIF EXISTS (SELECT * FROM productos_casi_agotados() AS prod_ago JOIN producto ON producto.codigo_barras = NEW.codigo_barras WHERE prod_ago.nombre = producto.nombre) THEN
RAISE NOTICE 'LOS PRODUCTOS YA CASI SE AGOTAN';
END IF;

UPDATE inventario_producto
SET cantidad = cantidad - NEW.cant_art
WHERE inventario_producto.codigo_barras = NEW.codigo_barras;
UPDATE venta
SET cant_Art_Total = cant_Art_Total + NEW.cant_art,
precio_Total = precio_Total + (precio_art[1]*NEW.cant_art),
fecha_venta = NOW()
WHERE venta.id_venta = NEW.id_venta;
RETURN NEW;
END;
$validar_compra$ LANGUAGE plpgsql;

CREATE TRIGGER validar_compra BEFORE INSERT OR UPDATE ON compra
FOR EACH ROW EXECUTE PROCEDURE validar_compra();
```

Figura 11: Se muestra la función para el segundo requerimiento, se debe decrementar el stock por la cantidad vendida del artículo, y mandar una advertencia..


```

CREATE FUNCTION venta_total_diaria(date) RETURNS TABLE(venta_total_diaria money) AS
$$
WITH ventas_diarias(venta_total_diaria, fecha_venta)
AS
(
    SELECT SUM(precio_total), fecha_venta
    FROM venta
    GROUP BY fecha_venta
)
SELECT venta_total_diaria
FROM ventas_diarias
WHERE ventas_diarias.fecha_venta = $1;
$$
LANGUAGE SQL;

```

Figura 12: Se muestra la función para el tercer requerimiento, retornar la cantidad vendida dada una fecha de inicio o fin.

```

CREATE FUNCTION productos_casi_agotados() RETURNS TABLE(nombre varchar(40)) AS
$$
SELECT producto.nombre
FROM producto
JOIN inventario_producto
ON producto.codigo_barras = inventario_producto.codigo_barras
WHERE inventario_producto.cantidad < 3;
$$
LANGUAGE SQL;

```

Figura 13: Se muestra la función para el cuarto requerimiento, retornar el nombre de aquellos productos los cuales hay 3 o menos en stock.

```

CREATE FUNCTION factura(text) RETURNS
TABLE(nombre varchar(40), tipo_articulo varchar(5), cant_art smallint, precio_venta money, precio_total_compra money, cant_art_total smallint, precio_total money, nomb
AS
$$
SELECT
producto.nombre,
producto.tipo_articulo,
compra.cant_art,
producto.precio_venta,
producto.precio_venta * compra.cant_art,
venta.cant_art_total,
venta.precio_total,
CONCAT(cliente.nombre, ' ', cliente.ap_pat, ' ', cliente.ap_mat)
FROM compra
JOIN venta
ON venta.id_venta = compra.id_venta
JOIN producto
ON compra.codigo_barras = producto.codigo_barras
JOIN cliente
ON venta.id_cliente = cliente.id_cliente
WHERE compra.id_venta = $1;
$$
LANGUAGE SQL;

```

Figura 14: Se muestra la función para el quinto requerimiento, retornar una vista que genere información asimilando una factura.

```

CREATE UNIQUE INDEX producto_index ON producto(codigo_barras ASC);

```

Figura 15: Se muestra la función para el sexto requerimiento, crear un índice, Se creo este índice para poder acceder de una forma muy eficiente a los registros de los productos.

6. CONCLUSIONES

Zamora Ramírez Oswaldo Rafael

Durante el desarrollo de este proyecto puse en práctica muchos conocimientos que he adquirido en el paso de las materias que he cursado durante la carrera desde cosas básicas de fundamentos de programación, estructuras de datos, programación orientada a objetos pero principalmente la materia de bases de datos, durante el desarrollo y avance del proyecto encontré muchos problemas uno en especial que me costó trabajo solucionar fue la conexión de una base de datos remota con la API que desarrollamos afortunadamente este inconveniente fue solucionado con la ayuda de mi equipo, creo que este proyecto fue de muy buena ayuda para mi crecimiento profesional porque además de aprender a trabajar en equipo en un ambiente un poco más real al que podré encontrar en el campo laboral, aprendí a utilizar herramientas de desarrollo como la usada en la aplicación Mobile, igualmente creo que todos los objetivos planteados en un inicio del proyecto fueron completados satisfactoriamente por todo el equipo.

Ramírez Viguera Alma Yeremay

La realización de este proyecto fue de mucha ayuda pues se asemeja a un proyecto que se nos presentaría en la vida laboral. En cuanto al diseño de la base de datos, realizar el análisis no fue complicado gracias a los ejercicios resueltos en la clase, la creación de tablas tampoco lo fue, pues ya había trabajado con eso en laboratorio, además con ayuda de mis compañeros conocí la variable “serial” y aprendí el funcionamiento de Android Studio, además con la investigación que se hizo se logró implementar el formato para las ventas. La aplicación móvil resultó un reto, pues nunca había desarrollado una, además de que no sabíamos cómo conectar la base de datos con la API, pero con la ayuda del equipo se pudo resolver y se decidió que sería buena idea dividirlo en backend y frontend, para que en caso de ser necesario y el cliente lo pidiera, cambiar la app por una página web sin tener que cambiar todo lo de la base de datos. Por lo tanto considero que el objetivo planteado se cumplió y los retos fueron solucionados correctamente.

Sánchez Manjarrez Andrew

La administración de bases de datos es una piedra angular medular de la organización, consulta y rastreo para las pequeñas y medianas empresas que dependen principalmente de la información y manejo de datos para una mejor gestión del inventario, obtención de estadísticas, venta y posicionamiento en el mercado, etc. Este proyecto es una ventana hacia los requerimientos de campo funcionales mínimos que requiere un DBA y ayuda a concentrar una razón instrumental al estudio de las bases de datos vistas durante el curso. El desarrollo del proyecto muestra un conjunto integral de conocimientos que al fundirse consiguen un so-

lo artefacto funcional; desarrollo web, normalización de bases, conocimientos de cliente-servidor, entre otros. Por experiencia y lecciones aprendidas, agradezco el conocimiento de mis compañeros de equipo, que me instruyeron en los antecedentes previamente mencionados para lograr una aplicación web que sintetizara mi aprendizaje del curso.

7. PRESENTACIÓN

7.1. Ejecución de API

Con ayuda del kit de desarrollo de Flutter y Android, además de la base en el diagrama de conexiones presentado, se procederá a mostrar su prueba simulada del retorno de datos. Con ayuda de Android Studio, se simuló el comportamiento de la aplicación móvil, y se hicieron pruebas de casos de uso para verificar que se retornan los datos correctos.

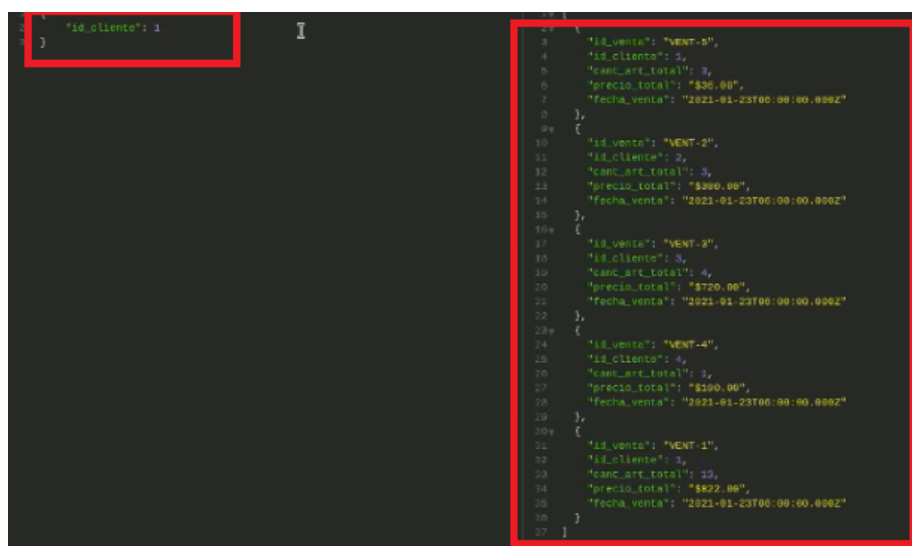


Figura 16: Se muestra la función de ejecución de la API en un cliente virtualizado. Se retornan los datos desde el servidor y se muestran los resultados en pantalla.

Se seleccionan los parámetros de visualización de la aplicación (Disposición, Medidas de Pantalla, entre otros) y los parámetros técnicos (Versión de SO, Métodos de Entrada y Salida, etc)

Se requiere de la imagen previa de un Android Virtual Device para configurar estos parámetros como se muestra en la figura 17.

7.2. Prueba de Aplicación

Se procedera a mostrar un caso de uso para la adición de un cliente nuevo.

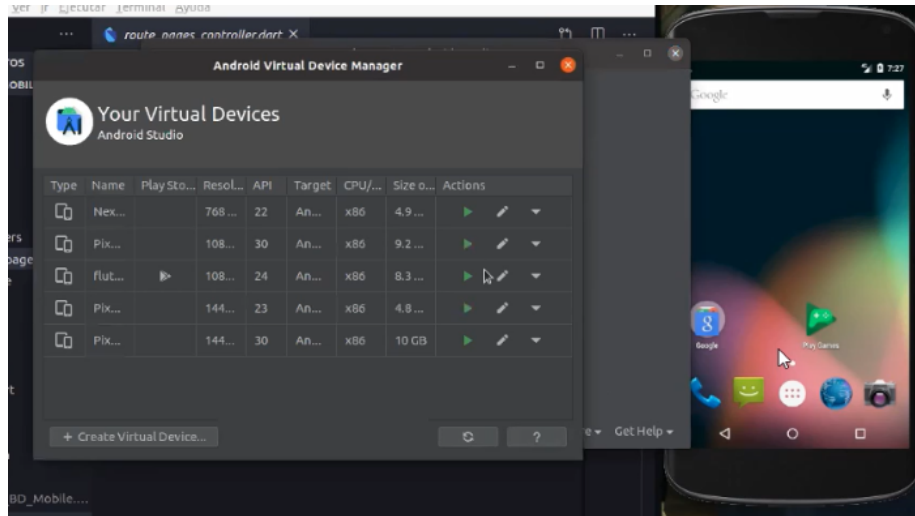


Figura 17: Ventana de Interfaz de Android Studio, con dispositivo seleccionado

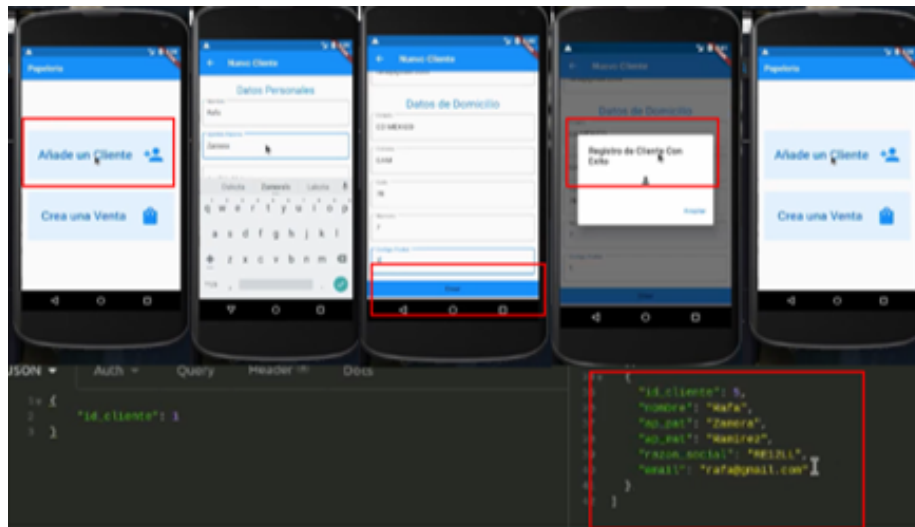


Figura 18: Se muestran las ventanas de interfaz de usuario para la creacion de un usuario, y la vista virtualizada de la respuesta del servidor terminado el proceso.