
Proyectos de computación aplicados a I.E.

Primer Parcial

David Antonio Rodas Alvarez

202010039

Fecha: Guatemala, 26 de agosto del 2024

Primer Examen Parcial*

David Antonio, Rodas Alvarez, 202010039^{1, **}

¹Escuela de Ingeniería Mecánica Eléctrica, Universidad de San Carlos, Guatemala.

(Dated: 27 de agosto de 2024)

I. RESUMEN

La quinta tarea del curso, el cual es un examen corto a realizar durante la hora de clase un programa en Octave que permita a una gasolinera llevar un registro de ventas, realizar el cobro y emitir facturas para sus clientes. El sistema debe ser capaz de registrar información sobre el tipo de combustible venido, la cantidad de litros despachados, el precio por litro, calcular el monto total a pagar, y emitir un recibo de factura. Toda la información registrada debe almacenarse en un archivo de texto y base de datos PostgreSQL para mantener un historial de transacciones. Requerimientos funcionales.

II. CÓDIGO

A. Octave

```
pkg load database;

% Funcin para conectarse a la base de datos PostgreSQL
function conn = connect_to_db()
    conn = pq_connect(setdbopts("dbname", "postgres", "host",
                                "localhost", "port", "5432", "user", "postgres",
                                "password", "202010039"));
end

% Funcin para seleccionar el tipo de combustible
function tipo_combustible = seleccionar_combustible()
    printf("\nSelecciona el tipo de combustible:\n");
    printf("1. Gasolina Regular\n");
    printf("2. Gasolina Sper\n");
    printf("3. Diesel\n");
    printf("4. Vpower\n");

    opcion = validar_entrada_numerica("Selecciona una opcin: ");

    switch opcion
        case 1
            tipo_combustible = "Gasolina Regular";
        case 2
            tipo_combustible = "Gasolina Sper";
        case 3
            tipo_combustible = "Diesel";
        case 4
            tipo_combustible = "Vpower";
        otherwise
            printf("Opcin no vlida. Selecciona una opcin vlida.\n");
            tipo_combustible = seleccionar_combustible(); % Volver a
                preguntar
    end
end

% Funcin para ingresar la cantidad de litros con validacin de
    entrada numrica
function cantidad_litros = ingresar_litros()
    cantidad_litros = validar_entrada_numerica("Ingresa la cantidad
        de litros a despachar: ");
    if cantidad_litros <= 0
        printf("La cantidad de litros debe ser un nmero positivo.
            Intenta de nuevo.\n");
        cantidad_litros = ingresar_litros(); % Volver a preguntar
    end
end
```

* Escuela de Ingeniería Mecánica Eléctrica
** e-mail: 3711111370101@ingenieria.usac.edu.gt

```
% Funcin para obtener el precio por litro segn el tipo de
    combustible
function precio_por_litro =
    obtenerPrecioCombustible(tipo_combustible)
    switch tipo_combustible
        case "Gasolina Regular"
            precio_por_litro = 3.50;
        case "Gasolina Sper"
            precio_por_litro = 4.00;
        case "Diesel"
            precio_por_litro = 3.00;
        case "Vpower"
            precio_por_litro = 4.50;
        otherwise
            precio_por_litro = 0; % En caso de error
    end
end

% Funcin para calcular el monto total a pagar
function monto_total = calcular_monto(cantidad_litros,
    precio_por_litro)
    monto_total = cantidad_litros * precio_por_litro;
end

% Funcin para mostrar el resumen de la transaccin
function mostrar_resumen(nombre_cliente, identificacion_vehiculo,
    tipo_combustible, cantidad_litros, precio_por_litro,
    monto_total)
    printf("\nResumen de la Transaccin:\n");
    printf("Nombre del cliente: %s\n", nombre_cliente);
    printf("Identificacin del vehculo: %s\n",
        identificacion_vehiculo);
    printf("Tipo de combustible: %s\n", tipo_combustible);
    printf("Cantidad de litros: %.2f\n", cantidad_litros);
    printf("Precio por litro: Q. %.2f\n", precio_por_litro);
    printf("Monto total a pagar: Q. %.2f\n", monto_total);
end

% Funcin para guardar la factura en la base de datos
function guardar_factura(conn, nombre_cliente,
    identificacion_vehiculo, tipo_combustible, cantidad_litros,
    precio_por_litro, monto_total)
    query = "INSERT INTO facturas (nombre_cliente,
        identificacion_vehiculo, tipo_combustible,
        cantidad_litros, precio_por_litro, monto_total) VALUES
        ($1, $2, $3, $4, $5, $6);";

    valores = {nombre_cliente, identificacion_vehiculo,
        tipo_combustible, cantidad_litros, precio_por_litro,
        monto_total};

    try
        pq_exec_params(conn, query, valores); % Ejecutar la consulta
            con parmetros
        printf("Factura registrada en la base de datos!\n");
    catch err
        error("Error ejecutando la consulta SQL: %s", err.message);
    end
end

% Funcin para generar la factura en el archivo de texto
```

```

function success = generar_factura_txt(nombre_cliente,
    identificacion_vehiculo, tipo_combustible, cantidad_litros,
    precio_por_litro, monto_total)
try
    fid = fopen("facturas.txt", "a");
    if fid == -1
        error("No se pudo abrir el archivo.");
    end

    fprintf(fid, "Nombre del cliente: %s\n", nombre_cliente);
    fprintf(fid, "Identificacin del vehculo: %s\n",
        identificacion_vehiculo);
    fprintf(fid, "Tipo de combustible: %s\n", tipo_combustible);
    fprintf(fid, "Cantidad de litros: %.2f\n", cantidad_litros);
    fprintf(fid, "Precio por litro: Q. %.2f\n", precio_por_litro);
    fprintf(fid, "Monto total a pagar: Q. %.2f\n", monto_total);
    fprintf(fid, "-----\n");

    fclose(fid);
    printf("Factura guardada en 'facturas.txt' con xito!\n");
    success = true;
catch
    printf("Error al escribir en el archivo 'facturas.txt'.\n");
    success = false;
end

% Funcin para validar que la entrada sea numrica
function valor = validar_entrada_numerica(mensaje)
while true
    valor = input(mensaje);
    if isnumeric(valor) && isscalar(valor)
        break;
    else
        printf("Entrada no vlida. Debes ingresar un nmero.\n");
    end
end
end

% Funcin para ver el historial de facturas desde la base de datos
% Funcin para ver el historial de facturas desde la base de datos
function ver_historial_facturas(conn)
query = "SELECT * FROM facturas;";
try
    result = pq_exec_params(conn, query, {}); % Ejecutar la
        consulta sin parmetros
    if isempty(result.data)
        printf("No hay facturas registradas an.\n");
    else
        printf("\nHistorial de Facturas:\n");
        for i = 1:rows(result.data)
            printf("Factura %d:\n", i);
            printf("Nombre del cliente: %s\n", result.data{i, 1});
            printf("Identificacin del vehculo: %s\n", result.data{i,
                2});
            printf("Tipo de combustible: %s\n", result.data{i, 3});
            printf("Cantidad de litros: %.2f\n", result.data{i, 4});
            printf("Precio por litro: Q. %.2f\n", result.data{i, 5});
            printf("Monto total: Q. %.2f\n", result.data{i, 6});
        end
    end
catch err
    error("Error ejecutando la consulta SQL: %s", err.message);
end
end

% Funcin para borrar el historial de facturas en la base de datos
function borrar_historial_facturas(conn)
confirmacion = input("Ests seguro de que deseas borrar todas
    las facturas? (s/n): ", "s");
if lower(confirmacion) == 's'
    query = "DELETE FROM facturas;";
    try
        pq_exec_params(conn, query, {}); % Ejecutar la consulta sin
            parmetros
        printf("Historial de facturas borrado con xito.\n");
    catch err
        error("Error ejecutando la consulta SQL: %s", err.message);
    end
else
    printf("Borrado cancelado.\n");
end
end

```

```

end

% Funcin principal con men
function main()
conn = connect_to_db(); % Conectar a la base de datos
PostgreSQL
while true
    clr; % Limpiar la pantalla
    printf("\nSistema de Gestin de Combustible\n");
    printf("1. Registrar transaccin\n");
    printf("2. Ver historial de facturas\n");
    printf("3. Borrar historial de facturas\n");
    printf("4. Salir\n");

    opcion = validar_entrada_numerica("Selecciona una opcin: ");

    switch opcion
        case 1
            % Registrar transaccin de combustible
            nombre_cliente = input("Por favor, ingresa el nombre del
                cliente: ", "s");
            identificacion_vehiculo = input("Ingresa la identificacin
                del vehculo (nmero de placa): ", "s");
            tipo_combustible = seleccionar_combustible();
            cantidad_litros = ingresar_litros();
            precio_por_litro =
                obtener_precio_combustible(tipo_combustible);
            monto_total = calcular_monto(cantidad_litros,
                precio_por_litro);
            mostrar_resumen(nombre_cliente, identificacion_vehiculo,
                tipo_combustible, cantidad_litros, precio_por_litro,
                monto_total);
            guardar_factura(conn, nombre_cliente,
                identificacion_vehiculo, tipo_combustible,
                cantidad_litros, precio_por_litro, monto_total);
            if ~generar_factura_txt(nombre_cliente,
                identificacion_vehiculo, tipo_combustible,
                cantidad_litros, precio_por_litro, monto_total)
                error("Error al escribir en el archivo 'facturas.txt'.
                    Programa terminado.");
            end
            printf("Transaccin registrada con xito.\n");
            pause(2); % Pausar para que el usuario pueda ver el
                mensaje antes de volver al men

        case 2
            % Ver historial de facturas
            ver_historial_facturas(conn);
            printf("\nPresiona Enter para regresar al men
                principal...\n");
            pause; % Espera a que el usuario presione Enter

        case 3
            % Borrar historial de facturas
            borrar_historial_facturas(conn);
            pause(2); % Pausar para que el usuario pueda ver el
                mensaje de confirmacin antes de volver al men

        case 4
            % Salir del programa
            printf("Saliendo del programa. Hasta pronto!\n");
            close(conn); % Cerrar la conexin a la base de datos
            break;

        otherwise
            % Manejar opcin no vlida
            printf("Opcin no vlida. Por favor, selecciona una opcin
                correcta.\n");
            pause(2); % Pausar para que el usuario pueda leer el
                mensaje de error
    end

    if opcion == 4
        break; % Salir del bucle si la opcin es 4
    end
end

% Ejecutar la funcin principal
main();

```

Figura 1: Creación de tabla en PostgreSQL

```
postgres=# CREATE TABLE facturas (
postgres(# id SERIAL PRIMARY KEY,
postgres(# identificacion_vehiculo VARCHAR(50) NOT NULL,
postgres(# tipo_combustible VARCHAR(50) NOT NULL,
postgres(# cantidad_litros FLOAT NOT NULL,
postgres(# precio_por_litro FLOAT NOT NULL,
postgres(# monto_total FLOAT NOT NULL
postgres(# );
CREATE TABLE
postgres=#
```

Fuente: Elaboración Propia, 2024.

Figura 2: Menu Principal

```
Sistema de Gestión de Combustible
1. Registrar transacción
2. Ver historial de facturas
3. Borrar historial de facturas
4. Salir
Selecciona una opción: █
```

Fuente: Elaboración Propia, 2024.

Figura 3: Muestra de datos almacenados - Facturas Emitida PostgreSQL

```
postgres=# SELECT * FROM facturas;
 nombre_cliente | identificacion_vehiculo | tipo_combustible | cantidad_litros | precio_por_litro | monto_total
-----
 David Rodas    | P69GQKB                 | Diesel           | 8               | 3               | 24
 Jose Alejandro | P114PPK                 | Gasolina S-per  | 5               | 4               | 20
 Maria Ines     | P742JDL                 | Gasolina Regular| 6               | 3.5             | 21
(3 rows)
```

Fuente: Elaboración Propia, 2024.

Figura 4: Lista - Archivo de texto (factura.txt)

```
-----
Nombre del cliente: Jose Alejandro
Identificación del vehículo: P114PPK
Tipo de combustible: Gasolina S&per
Cantidad de litros: 5.00
Precio por litro: 4.00
Monto total a pagar: 20.00
-----
Nombre del cliente: Maria Ines
Identificación del vehículo: P742JDL
Tipo de combustible: Gasolina Regular
Cantidad de litros: 6.00
Precio por litro: Q. 3.50
Monto total a pagar: Q. 21.00
-----
Nombre del cliente: python -u "d:\Desktop\Proyectos\1P_202010039_Py.py"
Identificación del vehículo: 3
Tipo de combustible: Gasolina Súper
Cantidad de litros: 3.00
Precio por litro: Q. 4.00
Monto total a pagar: Q. 12.00
-----
Nombre del cliente: Luis Pedro
Identificación del vehículo: C124KLZ
Tipo de combustible: Gasolina Súper
Cantidad de litros: 10.00
Precio por litro: Q. 4.00
Monto total a pagar: Q. 40.00
-----
```

Fuente: Elaboración Propia, 2024.

Figura 5: 4. Exit del programa

```
Ingrese la opción deseada: 4
Gracias por usar el programa!
```

Fuente: Elaboración Propia, 2024.

B. Python

```
pkg load database;

% Funcin para conectarse a la base de datos PostgreSQL
function conn = connect_to_db()
    conn = pq_connect(setdbopts("dbname", "postgres", "host",
                                "localhost", "port", "5432", "user", "postgres",
                                "password", "202010039"));
end

% Funcin para seleccionar el tipo de combustible
function tipo_combustible = seleccionar_combustible()
    printf("\nSelecciona el tipo de combustible:\n");
    printf("1. Gasolina Regular\n");
    printf("2. Gasolina Sper\n");
    printf("3. Diesel\n");
    printf("4. Vpower\n");

    opcion = validar_entrada_numerica("Selecciona una opcin: ");

    switch opcion
        case 1
            tipo_combustible = "Gasolina Regular";
        case 2
            tipo_combustible = "Gasolina Sper";
        case 3
            tipo_combustible = "Diesel";
        case 4
            tipo_combustible = "Vpower";
        otherwise
            printf("Opcin no vlida. Selecciona una opcin vlida.\n");
            tipo_combustible = seleccionar_combustible(); % Volver a
                preguntar
        end
    end

% Funcin para ingresar la cantidad de litros con validacin de
    entrada numrica
function cantidad_litros = ingresar_litros()
    cantidad_litros = validar_entrada_numerica("Ingresa la cantidad
        de litros a despachar: ");
    if cantidad_litros <= 0
        printf("La cantidad de litros debe ser un nmero positivo.
            Intenta de nuevo.\n");
        cantidad_litros = ingresar_litros(); % Volver a preguntar
    end
end

% Funcin para obtener el precio por litro segn el tipo de
    combustible
function precio_por_litro =
    obtenerPrecioCombustible(tipo_combustible)
    switch tipo_combustible
        case "Gasolina Regular"
            precio_por_litro = 3.50;
        case "Gasolina Sper"
            precio_por_litro = 4.00;
        case "Diesel"
            precio_por_litro = 3.00;
        case "Vpower"
            precio_por_litro = 4.50;
        otherwise
            precio_por_litro = 0; % En caso de error
    end
end

% Funcin para calcular el monto total a pagar
function monto_total = calcular_monto(cantidad_litros,
    precio_por_litro)
    monto_total = cantidad_litros * precio_por_litro;
end

% Funcin para mostrar el resumen de la transaccin
function mostrar_resumen(nombre_cliente, identificacion_vehiculo,
    tipo_combustible, cantidad_litros, precio_por_litro,
    monto_total)
    printf("\nResumen de la Transaccin:\n");
    printf("Nombre del cliente: %s\n", nombre_cliente);
    printf("Identificacin del vehiculo: %s\n",
        identificacion_vehiculo);
```

```

printf("Tipo de combustible: %s\n", tipo_combustible);
printf("Cantidad de litros: %.2f\n", cantidad_litros);
printf("Precio por litro: Q. %.2f\n", precio_por_litro);
printf("Monto total a pagar: Q. %.2f\n", monto_total);
end

% Funcin para guardar la factura en la base de datos
function guardar_factura(conn, nombre_cliente,
    identificacion_vehiculo, tipo_combustible, cantidad_litros,
    precio_por_litro, monto_total)
query = "INSERT INTO facturas (nombre_cliente,
    identificacion_vehiculo, tipo_combustible,
    cantidad_litros, precio_por_litro, monto_total) VALUES
    ($1, $2, $3, $4, $5, $6);";

valores = {nombre_cliente, identificacion_vehiculo,
    tipo_combustible, cantidad_litros, precio_por_litro,
    monto_total};

try
    pq_exec_params(conn, query, valores); % Ejecutar la consulta
    con parametros
    printf("Factura registrada en la base de datos!\n");
catch err
    error("Error ejecutando la consulta SQL: %s", err.message);
end

% Funcin para generar la factura en el archivo de texto
function success = generar_factura_txt(nombre_cliente,
    identificacion_vehiculo, tipo_combustible, cantidad_litros,
    precio_por_litro, monto_total)
try
    fid = fopen("facturas.txt", "a");
    if fid == -1
        error("No se pudo abrir el archivo.");
    end

    fprintf(fid, "Nombre del cliente: %s\n", nombre_cliente);
    fprintf(fid, "Identificacin del vehculo: %s\n",
        identificacion_vehiculo);
    fprintf(fid, "Tipo de combustible: %s\n", tipo_combustible);
    fprintf(fid, "Cantidad de litros: %.2f\n", cantidad_litros);
    fprintf(fid, "Precio por litro: Q. %.2f\n", precio_por_litro);
    fprintf(fid, "Monto total a pagar: Q. %.2f\n", monto_total);
    fprintf(fid, "-----\n");

    fclose(fid);
    printf("Factura guardada en 'facturas.txt' con xito!\n");
    success = true;
catch
    printf("Error al escribir en el archivo 'facturas.txt'.\n");
    success = false;
end

% Funcin para validar que la entrada sea numrica
function valor = validar_entrada_numerica(mensaje)
while true
    valor = input(mensaje);
    if isnumeric(valor) && isscalar(valor)
        break;
    else
        printf("Entrada no vlida. Debes ingresar un nmero.\n");
    end
end

% Funcin para ver el historial de facturas desde la base de datos
% Funcin para ver el historial de facturas desde la base de datos
function ver_historial_facturas(conn)
query = "SELECT * FROM facturas;";
try
    result = pq_exec_params(conn, query, {}); % Ejecutar la
    consulta sin parametros
    if isempty(result.data)
        printf("No hay facturas registradas an.\n");
    else
        printf("\nHistorial de Facturas:\n");
        for i = 1:rows(result.data)
            printf("Factura %d:\n", i);
            printf("Nombre del cliente: %s\n", result.data{i, 1});

```

```

        printf("Identificacin del vehculo: %s\n", result.data{i,
            2});
        printf("Tipo de combustible: %s\n", result.data{i, 3});
        printf("Cantidad de litros: %.2f\n", result.data{i, 4});
        printf("Precio por litro: Q. %.2f\n", result.data{i, 5});
        printf("Monto total: Q. %.2f\n", result.data{i, 6});
    end
end
catch err
    error("Error ejecutando la consulta SQL: %s", err.message);
end

% Funcin para borrar el historial de facturas en la base de datos
function borrar_historial_facturas(conn)
confirmacion = input("Ests seguro de que deseas borrar todas
    las facturas? (s/n): ", "s");
if lower(confirmacion) == 's'
    query = "DELETE FROM facturas;";
    try
        pq_exec_params(conn, query, {}); % Ejecutar la consulta sin
        parametros
        printf("Historial de facturas borrado con xito.\n");
    catch err
        error("Error ejecutando la consulta SQL: %s", err.message);
    end
else
    printf("Borrado cancelado.\n");
end

% Funcin principal con men
function main()
    conn = connect_to_db(); % Conectar a la base de datos
    PostgreSQL
    while true
        clr; % Limpiar la pantalla
        printf("\nSistema de Gestin de Combustible\n");
        printf("1. Registrar transaccin\n");
        printf("2. Ver historial de facturas\n");
        printf("3. Borrar historial de facturas\n");
        printf("4. Salir\n");

        opcion = validar_entrada_numerica("Selecciona una opcin: ");

        switch opcion
            case 1
                % Registrar transaccin de combustible
                nombre_cliente = input("Por favor, ingresa el nombre del
                    cliente: ", "s");
                identificacion_vehiculo = input("Ingresa la identificacin
                    del vehculo (nmero de placa): ", "s");
                tipo_combustible = seleccionar_combustible();
                cantidad_litros = ingresar_litros();
                precio_por_litro =
                    obtener_precio_combustible(tipo_combustible);
                monto_total = calcular_monto(cantidad_litros,
                    precio_por_litro);
                mostrar_resumen(nombre_cliente, identificacion_vehiculo,
                    tipo_combustible, cantidad_litros, precio_por_litro,
                    monto_total);
                guardar_factura(conn, nombre_cliente,
                    identificacion_vehiculo, tipo_combustible,
                    cantidad_litros, precio_por_litro, monto_total);
            if ~generar_factura_txt(nombre_cliente,
                identificacion_vehiculo, tipo_combustible,
                cantidad_litros, precio_por_litro, monto_total)
                error("Error al escribir en el archivo 'facturas.txt'.
                    Programa terminado.");
            end
            printf("Transaccin registrada con xito.\n");
            pause(2); % Pausar para que el usuario pueda ver el
                mensaje antes de volver al men

            case 2
                % Ver historial de facturas
                ver_historial_facturas(conn);
                printf("\nPresiona Enter para regresar al men
                    principal...\n");
                pause; % Espera a que el usuario presione Enter

            case 3
                % Borrar historial de facturas

```

```

    borrar_historial_facturas(conn);
    pause(2); % Pausar para que el usuario pueda ver el
        mensaje de confirmacin antes de volver al men

case 4
    % Salir del programa
    printf("Saliendo del programa. Hasta pronto!\n");
    close(conn); % Cerrar la conexin a la base de datos
    break;

otherwise
    % Manejar opcin no vlida
    printf("Opcin no vlida. Por favor, selecciona una opcin
        correcta.\n");
    pause(2); % Pausar para que el usuario pueda leer el
        mensaje de error
end

if opcion == 4
    break; % Salir del bucle si la opcin es 4
end
end
end

% Ejecutar la funcin principal
main();

```

Figura 6: Menu Principal

```

Sistema de Gestión de Combustible
1. Registrar transacción
2. Ver historial de facturas
3. Borrar historial de facturas
4. Salir
Selecciona una opción: █

```

Fuente: Elaboración Propia, 2024.

Figura 7: Ingreso de datos

```

Selecciona una opción: 1
Por favor, ingrese el nombre del cliente: Juana Santos
Ingrese la identificación del vehículo (numero de placa): 0451QKV
Selecciona el tipo de combustible:
1. Gasolina Súper
2. Gasolina Regular
3. Diesel
4. Otros
Presione la tecla Enter para continuar: █

```

Fuente: Elaboración Propia, 2024.

Figura 8: Resumen de transacción

```

Resumen de la Transacción:
Nombre del cliente: Juana Santos
Identificación del vehículo: 0451QKV
Tipo de combustible: Gasolina Súper
Cantidad de litros: 5.00
Precio por litro: Q. 4.00
Monto total a pagar: Q. 20.00
¡Factura registrada en la base de datos!
¡Factura guardada en 'facturas.txt' con éxito!
Transacción registrada con éxito.
Presione Enter para continuar... █

```

Fuente: Elaboración Propia, 2024.

Figura 9: 4. Exit del programa

```

Ingrese la opción deseada: 4
Gracias por usar el programa!

```

Fuente: Elaboración Propia, 2024.

C. Link de GitHub

<https://github.com/dvd-r16/proyectos>

III. CONCLUSIÓN

1. Fue posible realizar el programa durante el horario de clases. El circuito del programa realiza todo lo que la gasolinera pidió. Además, ofrece funcionalidades para guardar los resultados en un archivo de texto, leer la información almacenada previamente y borrar el archivo en caso de ser necesario. El programa fue implementado de manera que permite al usuario interactuar de forma sencilla con las diferentes opciones, asegurando que el flujo de ejecución sea claro y eficiente, proporcionando al usuario una herramienta funcional para la gestión de sus datos de salud de forma automatizada.

[1] Ing. Jose Anibal Silva de los Angeles. Proyectos Aplicados para I.E. *Programa del curso*. Ciudad de Guatemala:

Universidad de San Carlos, Facultad de Ingeniería, Escuela de Ingeniería Mecánica Eléctrica.
