HITO 2 DEL 1º TRIMESTRE DE PROGRAMACIÓN

Alejandro Cortés Díaz

CampusFP

Diseño de aplicaciones multiplataforma – programación  20/11/2024

Índice

[**DESARROLLO** 2](#_Toc182245322)

[**Antes de crear el CV, lo primero que te solicitan es un trabajo de investigación académico por escrito sobre los puntos que se exponen a continuación y plasmarlo de tal forma que seas capaz de identificar fácilmente los temas y los contenidos asociados a esos temas. Se** 2](#_Toc182245323)

[**recomienda tomar referencias de otras páginas web con temática similar. Se valorará el uso de elementos multimedia. Debéis consultar varias fuentes cuya veracidad sea demostrable y explicar con vuestras propias palabras cada uno de los apartados, incluyendo información adicional y complementaria sobre el enunciado del mismo. Debéis determinar una correcta estructura de las categorías más importantes en el apartado clasificación.** 2](#_Toc182245324)

[**Debe contener:** 3](#_Toc182245325)

[**1.** **Explica los orígenes de los lenguajes de marcas.** 3](#_Toc182245326)

[**2.** **Explicar los diferentes estándares y organismos oficiales en el desarrollo y diseño de lenguajes de marcas.** 4](#_Toc182245327)

[**3.** **Explica los orígenes y las diferencias entre CSS y SCSS.** 6](#_Toc182245328)

[**4.** **Principales etiquetas HTML5, así como una breve descripción de que son y cómo funcionan las etiquetas semánticas.** 8](#_Toc182245329)

[**5.** **Principales elementos CSS y su aplicación.** 11](#_Toc182245330)

[**6. Guía de creación de un sitio web con el paso a paso del desarrollo.** 13](#_Toc182245331)

[**7.** **Guia y uso de librerías de CSS para el desarrollo web** 15](#_Toc182245332)

[**Enlace a GitHub** 18](#_Toc182245333)

[**Bibliografía** 19](#_Toc182245334)

# **Archivo donde ejecuto;**

# IMPORTO CONEXION

import Conexion as Cx

# IMPORTO MAIN MENU

import Menu\_Main as Mm

# ASIGNO CONEXION A CONEXION PARA UTILIZAR SUS FUNCIONES CON LA BASE DE DATOS

conexion = Cx.conexion

# ASIGNO CURSOR A CURSOR PARA HACER EXECUTES

cursor = Cx.cursor

# LLAMO A MENU MAIN PARA PODER USAR CURSOR Y CONEXION EN EL

Mm.Menu(conexion, cursor)

# **Menu conexión padre:**

import mysql.connector

def conectar (BaseDatos):

 # Establecer conexión con la base de datos

        conexion = mysql.connector.connect(

            host="localhost",       # Dirección del servidor (localhost para base de datos local)

            user="root",         # Usuario de la base de datos

            password="curso",  # Contraseña del usuario

            database=BaseDatos    # Nombre de la base de datos

        )

        return conexion

Menu conexión hijo;

#IMPORTO DESDE CONEXIÓN PADRE LAS FUNCIONES DE CONEXION

import Conexion\_Padre as bdd

#USO CONEXIÓN PARA CONECTARME A BD

conexion=bdd.conectar("SupermercadoPython")

#EMPLEO CURSOR PARA EJECUTAR

cursor= conexion.cursor()

Menu main;

#IMPORTO LOS DOS MENUS PARA ENLAZARLOS AL MAIN

import Menu\_Producto as Mp

import Menu\_Cliente as Mc

#DEF MENU

def Menu (conexion,cursor):

    try:

        while True:

            #MUESTRO OPCIONES Y ASIGNO CADA UNA A CADA CASO POSIBLE

            opcion=int(input("Menú de opciones\n1/Tabla Producto\n2/Tabla Cliente\n4/Salir\n"))

            match opcion:

                case 1:

                    print("Marchando al menu de Producto")

                    Mp.Menu(conexion,cursor)

                case 2:

                    print("Marchando al menu de Cliente")

                    Mc.Menu(conexion,cursor)

                case 3:

                    print("Saliendo del programa")

                    cursor.close()

                    conexion.close()

                    break

    except ValueError as e:

        #ERROR SI TIPO DE DATO ERRONEO

        print("No aplica ese valor, error{e}")

    except Exception:

        #ERROR SI FALLA ALGUNA COSA DISTINTA

        print("Error al ejecutar la funcion")

Menu cliente;

# IMPORTO CLIENTE PARA USARLO EN EL MENU

import DAW\_Cliente as dwC

# DEF MENU

def Menu(conexion, cursor):

    try:

        while True:

            # MUESTRO OPCIONES

            opcion = int(input("Menú de opciones \n1/Insertar nuevo cliente\n2/Seleccionar todos los clientes\n3/Seleccionar un cliente en específico\n4/Modificar cliente\n5/Seguimiento de los pedidos \n6/Salir al menu principal"))

            # ASIGNO UNA OPCION A CADA FUNCION CON MATCH CASE CORRESPONDIENTE

            match opcion:

                case 1:

                    print("Insertar cliente")

                    dwC.ingresar\_cliente(conexion, cursor)

                case 2:

                    print("Seleccionar a todos los clientes")

                    dwC.select\_clientes(cursor)

                case 3:

                    print("Seleccionar cliente")

                    dwC.select\_cliente(cursor)

                case 4:

                    print("Modificar cliente")

                    dwC.modify\_cliente(conexion, cursor)

                case 5:

                    print("Seguimiento de cliente")

                    dwC.seguimiento\_cliente(cursor)

                case 6:

                    print("Marchando al menú principal")

                    break

    #ERROR SI TIPO DE DATO NO ES CORRECTO

    except ValueError as e:

        print(f"No aplica ese valor, error {e}")

    except Exception:

        #ERROR SI SE DA OTRO TIPO DE FALLO

        print("Error al ejecutar la función")

Menu producto;

# IMPORTO DAW PRODUCTO PARA USARLO EN LOS DISTINTOS CASES

import DAW\_Producto as dwp

# DEFINO MENU

def Menu(conexion, cursor):

    try:

        while True:

            # OPCIONES

            opcion = int(input("Menú de opciones \n1/Comprar producto\n2/Seleccionar todos los productos\n3/Seleccionar un producto en particular\n5/Salir"))

            # SEGUN LA OPCION UNA FUNCION PARA LA NECESIDAD CORRESPONDIENTE

            match opcion:

                case 1:

                    print("Comprar producto")

                    dwp.compra\_producto(conexion, cursor)

                case 2:

                    print("Seleccionar todos los productos")

                    dwp.select\_productos(cursor)

                case 3:

                    print("Seleccionar producto")

                    dwp.select\_producto(cursor)

                case 4:

                    print("Marchando al menu principal")

                    break

    except ValueError as e:

        # ERROR SI EL TIPO DE DATO NO ES VALIDO

        print(f"No aplica ese valor, error {e}")

    except Exception:

        # ERROR PARA CUALQUIER OTRO TIPO DE FALLO

        print("Error al ejecutar la funcion")

DAW CLIENTE;

#DEF FUNCION PARA INTRODUCIR CLIENTE A BASE DE DATOS

def ingresar\_cliente(conexion, cursor):

    try:

        # SOLICITO ATRIBUTOS DE LA TABLA CLIENTE

        idcliente = int(input("Seleccione su id: "))

        Nombre = input("Ingrese su nombre: ")

        ciudad = input("Dime la ciudad del nuevo cliente: ")

        tlf = input("Ingrese su número de teléfono: ")

        # LOS INSERTO EN SENTENCIA SQL Y EJECUTO

        consulta = "INSERT INTO cliente(idcliente, Nombre, ciudad, tlf) VALUES (%s, %s, %s, %s)"

        cursor.execute(consulta, (idcliente, Nombre, ciudad, tlf))

        conexion.commit()

        # GUARDO LA INSERCION EN UN ARRAY, Y ESTE LO PRINTEO EN UN ARCHIVO

        array = [idcliente, Nombre, ciudad, tlf]

        with open("Clientes.txt", "a") as archivo:

            archivo.write("=== Lista de Clientes ===\n")

            archivo.write("Datos insertados:\n\n")

            for dato in array:

                archivo.write(f"{dato}\n")

        print(f"Se insertó en la base de datos el cliente: {Nombre} de la ciudad: {ciudad} con el número {tlf}")

        #LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS POSIBLES)

    except ValueError as e:

        print(f"Tipo de dato incorrecto, error {e}")

    except Exception as unknown:

        print(f"No pudo ejecutarse la función, error {unknown}")

# DEF FUNCION PARA SELECCIONAR TODOS LOS CLIENTES RECOGIDOS (SOLO HACE FALTA CURSOR PORQUE ES SENTENCIA DE TIPO SELECT)

def select\_clientes(cursor):

    try:

        # CONSULTO A TODOS LOS CLIENTES DE LA TABLA ORDENANDOLOS POR EL ID A TRAVES DE SENTENCIA SQL

        consulta = "SELECT \* FROM cliente ORDER BY idcliente"

        cursor.execute(consulta)

        # PRINTEO A LOS CLIENTES RECOGIDOS

        resultado = cursor.fetchall()

        for fila in resultado:

            print(f"IDCliente: {fila[0]}, Nombre: {fila[1]}, Ciudad: {fila[2]}, Teléfono: {fila[3]}")

        #LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS POSIBLES)

    except ValueError as e:

        print(f"Tipo de dato incorrecto, error {e}")

    except Exception as unknown:

        print(f"No pudo ejecutarse la función, error {unknown}")

# FUNCION PARA SELECCIONAR CLIENTE INDIVIDUAL (SENTENCIA SELECT POR LO QUE NO HACE FALTA CONEXION)

def select\_cliente(cursor):

    try:

        # PIDO ID PARA SACAR SUS DATOS

        idcliente = int(input("Dime el ID del cliente que quieres seleccionar: "))

        # EXTRAIGO CADA DATO DEL CLIENTE CON EL ID DADO

        consulta = "SELECT \* FROM cliente WHERE idcliente = %s ORDER BY idcliente"

        cursor.execute(consulta, (idcliente,))

        # LOS IMPRIMO

        resultado = cursor.fetchall()

        for fila in resultado:

            print(f"IDCliente: {fila[0]}, Nombre: {fila[1]}, Ciudad: {fila[2]}, Teléfono: {fila[3]}")

        #LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS POSIBLES)

    except ValueError as e:

        print(f"Tipo de dato incorrecto, error {e}")

    except Exception as unknown:

        print(f"No pudo ejecutarse la función, error {unknown}")

# DEF FUNCION PARA MOSTRAR LOS DATOS DE LAS COMPRAS O PEDIDOS DE UN SOLO CLIENTE

def seguimiento\_cliente(cursor):

    try:

        # PIDO SU ID

        idcliente = int(input("Introduce el ID del cliente para realizar un seguimiento: "))

        # INNER JOINS DE LAS TABLAS DE LAS QUE SELECCIONO TODOS LOS DATOS SOBRE EL CLIENTE Y SUS PEDIDOS

        consulta = """

        SELECT c.idcliente, c.Nombre, c.ciudad, c.tlf, p.idpedido, prod.nombre, d.precio, d.unidades

        FROM cliente c

        JOIN pedido p ON c.idcliente = p.idcliente

        JOIN detalle d ON d.idpedido = p.idpedido

        JOIN producto prod ON prod.idproducto = d.idproducto

        WHERE c.idcliente = %s

        ORDER BY p.idpedido

        """

        #EJECUTO

        cursor.execute(consulta, (idcliente,))

        # LOS PRINTEA

        resultado = cursor.fetchall()

        for fila in resultado:

            print(f"ID Cliente: {fila[0]}")

            print(f"Nombre: {fila[1]}")

            print(f"Ciudad: {fila[2]}")

            print(f"Teléfono: {fila[3]}")

            print(f"Pedido ID: {fila[4]}")

            print(f"Producto: {fila[5]}, Precio: {fila[6]}, Cantidad: {fila[7]}\n")

        #LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS POSIBLES)

    except ValueError as e:

        print(f"Error en la entrada de datos: {e}")

    except Exception as unknown:

        print(f"No pudo ejecutarse la función, error: {unknown}")

DAW producto;

#DEF FUNNCION PARA LA COMPRA DE PRODUCTO

def compra\_producto (conexion,cursor):

    try:

        while True:

            try:

                #PRIMERO ME ASEGURO DE QUE EXISTA EL CLIENTE QUE VA  A COMPRAR

                cliente=input("Se encuentra en la base de datos? Si/No")

                if cliente.lower() == "si":

                    print("Inicio de sesión")

                    idcliente=int(input("Introduzca su id de cliente"))

                    #SELECCIONO HIPOTETICO CLIENTE CON LA ID QUE ME DAN

                    consulta = "SELECT \* FROM cliente WHERE idcliente = %s"

                    cursor.execute(consulta, (idcliente,))

                    resultado = cursor.fetchone()

                    #SI SE PUEDE REALIZAR LA CONSULTA, LUEGO EXISTE CLIENTE

                    if resultado:

                        print(f"Bienvenido!")

                        break

                    #SINO,

                    else:

                        print("El ID de cliente no existe en la base de datos.")

                #PERMITO REGISTRO SI NO ESTA REGISTRADO EN LA DB

                else:

                    idcliente=int(input("Ingrese su idcliente"))

                    Nombre=input("Ingrese su nombre: ")

                    ciudad=(input("Ingrese su ciudad: "))

                    tlf=input("Ingrese su telefono")

                    #EJECUTO CONSULTA CON LOS DATOS DADOS

                    consulta="insert into cliente(idcliente,Nombre,ciudad,tlf) values(%s,%s,%s,%s)"

                    cursor.execute(consulta,(idcliente,Nombre,ciudad,tlf))

                    conexion.commit()

                    #GUARDO AL CLIENTE EN EL TXT COMO ARRAY UNA VEZ SE REGISTRE

                    array = [idcliente, Nombre, ciudad, tlf]

                    with open("Clientes.txt", "a") as archivo:

                        archivo.write("=== Lista de Clientes ===\n")

                        archivo.write("Datos insertados:\n\n")

                        for dato in array:

                            archivo.write(f"{dato}\n")

                    print(f"Se insertó el cliente: {Nombre}\n con el id: {idcliente}\n la ciudad: {ciudad}\n y el telefono {tlf}")

                    break

             #LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS POSIBLES)

            except ValueError as e:

                print(f"Tipo de dato incorrecto, error {e}")

            except Exception as unknown:

                print(f"No pudo ejecutarse la función, error {unknown}")

        while True:

            try:

                idcliente=int(input("Cual es su id de cliente?\n"))

                lista= "select idproducto,Nombre,precio from producto"

                cursor.execute(lista)

                resultado=cursor.fetchall()

                for idproducto, nombre,precio in resultado:

                    print(f"IDProducto: {idproducto}, Nombre: {nombre}, Precio: {precio}")

                fechapedido= input("En qué fecha realizó este pedido")

                fechaentrega= input("Qué día la recibe")

                #INSERT INTO EN PEDIDO ANTES DE COMPRA

                consulta1=("insert into pedido(idcliente,fechapedido,fechaentrega) values(%s,%s,%s)")

                cursor.execute(consulta1,(idcliente,fechapedido,fechaentrega))

                conexion.commit()

                #INTRODUCCIÓN DE DATOS NECESARIOS PARA ASIGNARLOS A LOS REGISTROS DE LA TABLA DETALLE

                unidades = int(input("Cuántas unidades deseas comprar del producto al que echo un ojo?"))

                idproducto = int(input("De nuestros productos disponibles, introduce el id de aquel que quiera comprar:\n"))

                # OBTENCIÓN DEL PRECIO DEL PRODUCTO SELECCIONADO

                consulta2 = "select precio from producto where idproducto = %s"

                cursor.execute(consulta2, (idproducto,))

                resultado = cursor.fetchall()

                precio\_compra = resultado[0][0]

                #INSERT INTO EN DETALLE PARA COMPRA

                consulta3=("insert into detalle(idproducto,precio,unidades) values(%s,%s,%s)")

                cursor.execute(consulta3,(idproducto,precio\_compra,unidades))

                conexion.commit()

                print(f"Se realizó la compra de: {idproducto} con el coste: {precio\_compra}, y la cantidad: {unidades}")

                array=[f"El cliente con el id:{idcliente}Y el nombre:{Nombre},Realizo la compra del producto con el id:{idproducto} ,A un precio de:{precio\_compra} En la cantidad de :{unidades}"]

                #ARCHIVO TXT PARA INTRODUCIR LOS DATOS DE COMPRA

                with open("Factura.txt", "a") as archivo:

                    archivo.write("=== Factura de Compra ===\n")

                    archivo.write("Detalles de la compra:\n\n")

                    for dato in array:

                        archivo.write(f"{dato}\n")

                #PREGUNTO SI DESEA CONTINUAR CON LA COMPRA

                continuar=input("Deseas seguir comprando? Si/No\n")

                if continuar.lower() =="si":

                    print("Siguiente compra")

                elif continuar.lower() =="no":

                    break

            #LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS POSIBLES)

            except ValueError as e:

                print(f"Tipo de dato incorrecto, error {e}")

            except Exception as unknown:

                print(f"No pudo ejecutarse la función, error {unknown}")

    #LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS POSIBLES)

    except Exception as e:

        print(f"Error en la función 'compra\_producto': {e}")

    except Exception as unknown:

        print(f"No pudo ejecutarse la función, error {unknown}")

#DEF FUNCION PARA SELECCIONAR TODOS LOS PRODUCTOS

def select\_productos (cursor):

    try:

        #EJECUTO SELECT ALL DE PRODUCTOS

        consulta="select \* from producto order by idproducto"

        cursor.execute(consulta)

        #PRINTEA LOS RESULTADOS DE LA CONSULTA

        resultado=cursor.fetchall()

        for fila in resultado:

            print(f"IDProducto: {fila[0]}, Nombre: {fila[1]},  Precio: {fila[2]},")

    #LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS POSIBLES)

    except ValueError as e:

        print(f"Tipo de dato incorrecto, error {e}")

    except Exception as unkown:

        print(f"No pudo ejecutarse la funcion, error {unkown}")

#DEF FUNCION PARA SELECCIONAR UNO SOLO

def select\_producto (cursor):

    try:

        #EJECUTO SELECT PARA ENSEÑAR TODOS LOS PRODUCTOS Y POSTERIORMENTE ELEGIR Y EJECUTO

        lista= "select idproducto,nombre,from producto"

        cursor.execute(lista)

        #PRINTEO RESULTADOS

        resultado=cursor.fetchall()

        for idproducto, nombre,precio in resultado:

            print(f"IDProducto: {idproducto}, Nombre: {nombre}")

        #PREGUNTO EL ID DEL PRODUCTO DEL QUE QUIERE SABER EL PRECIO Y EJECUTO CON EL DATO

        idproducto=int(input("Que id tiene el producto del que deseas informacion?"))

        consulta="select \* from producto where idproducto=%s"

        cursor.execute(consulta,(idproducto,))

        #PRINTEO LOS RESULTADOS ASOCIADOS A LA CONSULTA SQL PREVIA

        resultado=cursor.fetchall()

        for idproducto, nombre,precio in resultado:

            print(f"IDProducto: {idproducto}, Nombre: {nombre}, Precio: {precio}")

    #LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS POSIBLES)

    except ValueError as e:

        print(f"Tipo de dato incorrecto, error {e}")

    except Exception as unkown:

        print(f"No pudo ejecutarse la funcion, error {unkown}")

# **Enlace a GitHub**

<https://github.com/Cortes-cmd/LenguajeDeMarcas.git>

# **Bibliografía**

ChatGPT. (s/f). Chatgpt.com. Recuperado el 11 de noviembre de 2024, de <https://chatgpt.com/c/67320241-fe88-8001-8566-c8d404dbf742>

*¿Conoces cuáles son las propiedades mas utilizadas en CSS?* (s/f). Mgpanel.org. Recuperado el 11 de noviembre de 2024, de <https://blog.mgpanel.org/post/-conoces-cuales-son-las-propiedades-mas-utilizadas-en-css->

*CSS básico*. (s/f). MDN Web Docs. Recuperado el 11 de noviembre de 2024, de <https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/CSS_basics>

*Desarrollo de Aplicaciones WEB. Introducción a HTML y CSS. Página Rafael Barzanallana. Universidad de Murcia*. (s/f). Www.um.es. Recuperado el 11 de noviembre de 2024, de <https://www.um.es/docencia/barzana/DAWEB/2017-18/daweb-tema-1-introduccion-html-css.html>

Hernandez, I. (2023, junio 22). *16 Frameworks Populares de CSS, Útiles Para Ahorrar Tiempo (Con Estilo)*. Guías para Sitios Web, Tips & Conocimiento; DreamHost. <https://www.dreamhost.com/blog/es/frameworks-css-populares/>

Islas, D. S. (2023, julio 10). Cómo crear una página web: Guía paso a paso. *Blog de Wix*. <https://es.wix.com/blog/guia-para-crear-paginas-web>

*¡Ohh, la UI!, ¿librerías de CSS o librerías de componentes, cuál usar?* (s/f). Fixtergeek.com. Recuperado el 11 de noviembre de 2024, de <https://fixtergeek.com/blog/ohh-la-ui-librerias-de-css-o-librerias-de-componentes-cual-usar>

Pavlik, V. (2023, junio 14). *HTML Semántico: Qué Es y Cómo Usarlo Correctamente*. Semrush Blog; Semrush. <https://es.semrush.com/blog/html-semantico/>

(S/f). Iebschool.com. Recuperado el 11 de noviembre de 2024, de <https://www.iebschool.com/blog/que-es-etiqueta-html-analitica-usabilidad/>