

MONAT (BANCO DE IMÁGENES)

DANNER RODRIGO MAMANI COPA

INTRODUCCIÓN

‘MONAT’ es una aplicación dedicada al contenido multimedia. Esta aplicación pretende ayudar a distintas áreas en las cuales se necesite contenido audiovisual para la creación de diversos proyectos. Hoy en día el copyright es una cosa bastante seria por lo que Monat protege a diversos usuarios de esta restricción.

La aplicación y su contenido es totalmente gratuita, pero como se financiara esta aplicación, pues con la afiliación con distintas tiendas online como amazon, esto por que la mayoría de usuarios de una aplicación como pinterest buscan imágenes para referenciar un producto que quieren comprar, entonces Monat sería un puente de comercio y contenido.

Como ya mencionamos anteriormente Monat es una aplicación para los estudiantes universitarios y distintos proyectos que se pueden presentar que requieran de contenido audiovisual sin tener que temerle a copyright.

ANÁLISIS Y DISEÑO

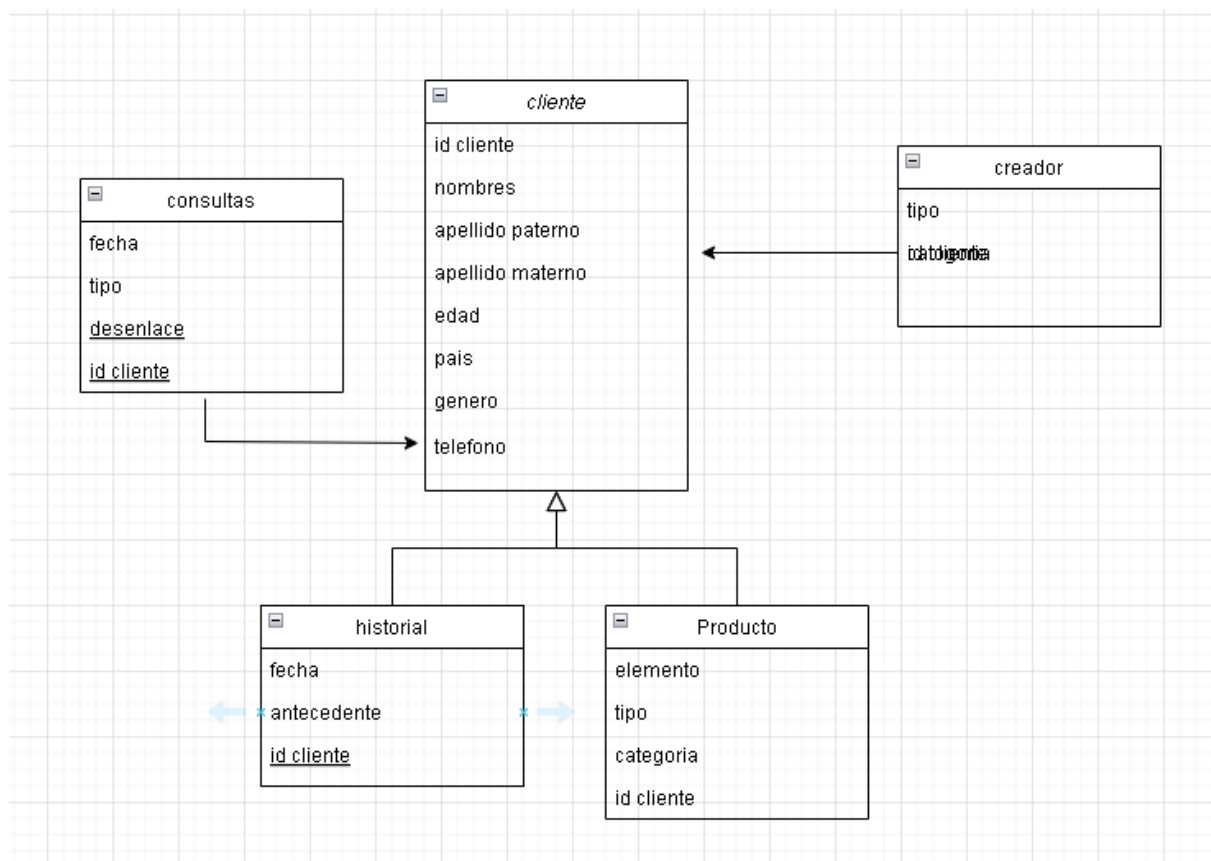
Para este proyecto se precisa lógicamente una base de datos el cual almacena información como: datos de usuario, datos de todo el contenido, un historial de la actividad, todas las descargas, y demás datos que son de mucha importancia.

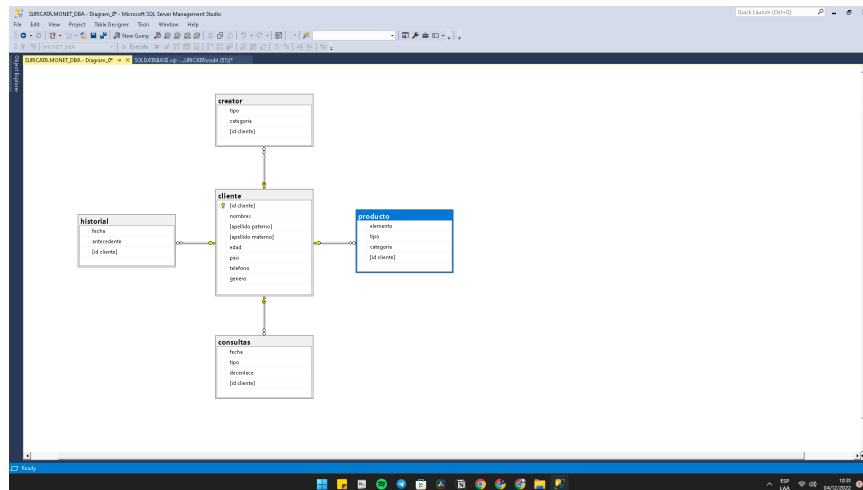
La base de datos se llama [MONET_DBA] el cuál hace referencia al nombre de la aplicación. También éste cuenta con 5 tablas las cuales son las que almacenan en forma ordenada toda la información relevante para la aplicación.

TABLA	DESCRIPCION
cliente	Almacena todos los datos de los clientes.
historial	Guarda el historial de un usuario.
producto	Tiene registro de todos los recursos en la aplicación.
descarga	Almacena las descargas del usuario.
consulta	Guarda información sobre el movimiento de un usuario luego de la búsqueda.

DISEÑO DE LA BASE DE DATOS

[ENLACE DEL DIAGRAMA \(CLIK\)](#)





```
CREATE DATABASE [MONET_DBA];
USE [MONET_DBA];
```

```
CREATE TABLE cliente(
    [id cliente] int primary key not null,
    [nombres] varchar(50) not null,
    [apellido paterno] varchar(50) not null,
    [apellido materno] varchar(50) not null,
    [edad] int not null,
    [pais] varchar(50) not null,
    [telefono] int not null,
    [genero] varchar(50) not null
);

CREATE TABLE producto(
    [elemento] int identity not null,
    [tipo] varchar(30) not null,
    [categoria] varchar(30) not null,
    [id cliente] int,
    FOREIGN KEY ([id cliente]) REFERENCES cliente([id cliente])
);

CREATE TABLE consultas(
    [fecha] varchar(30),
    [tipo] varchar(100),
    [decenlace] varchar(10),
    [id cliente] int,
    FOREIGN KEY ([id cliente]) REFERENCES cliente([id cliente])
);

CREATE TABLE historial(
    [fecha] varchar(20),
    [antecedente] varchar(50),
    [id cliente] int,
    FOREIGN KEY ([id cliente]) REFERENCES cliente ([id cliente])
);

CREATE TABLE creator(
    [tipo] varchar(30) not null,
    [categoria] varchar(30) not null,
    [id cliente] int,
    FOREIGN KEY ([id cliente]) REFERENCES cliente([id cliente])
);
```

}-----CONSULTAS SQL UTILIZANDO INNER JOIN-----

--determinar cuantos clientes creadores son del pais de argentina

```
SELECT COUNT([nombres]) as clientes
FROM cliente as cli
INNER JOIN creator as cre on cre.[id cliente] = cli.[id cliente]
WHERE pais = 'Argentina';
```

--determinar cuantos videos hay en la plataforma y a que clientes pertenecen

```
SELECT COUNT([nombres]) as clientes
FROM cliente as cli
INNER JOIN producto as pro on pro.[id cliente] = cli.[id cliente]
WHERE tipo = 'video';
```

--determinar cuantas descargas hubo en total que terminaron en descarga

```
SELECT COUNT([id cliente]) as descargas
FROM cliente as cli
INNER JOIN consultas as con on con.[id cliente] = cli.[id cliente]
WHERE [tipo] LIKE 'descarga$';
```

-- determinar cuantos productos son de la categoria entretenimiento que su creadores sean de Colombia

```
SELECT COUNT([categoria]) as clientes
FROM producto as pro
INNER JOIN cliente as cli on cli.[id cliente] = pro.[id cliente]
WHERE [pais] = 'Colombia'
and [categoria] = 'entrettenimiento';
```

}-----FUNCIONES SQL-----

--CREAR UNA FUNCION QUE CLASIFIQUE TODOS EL CONTENIDO DE LA PLATAFORMA

```
create function clasificacion(@tipo varchar(10))
returns varchar(100) as
begin
    declare @res VARCHAR(20);
    if (@tipo = 'gracioso')
        set @res = 'entretenimiento'
    if (@tipo = 'curso')
        set @res = 'educativo'
    if (@tipo = 'publicidad')
        set @res = 'comercial'
    if (@tipo = 'comun')
        set @res = 'recurso';
    return @res;
end;

select dbo.clasificacion('gracioso') as DBA;
select dbo.clasificacion('curso') as DBA;
select dbo.clasificacion('publicidad') as DBA;
select dbo.clasificacion('comun') as DBA;
```

-- CREAR UNA FUNCION QUE PERMITA MOSTRAR SI ES MAYOR A CIERTA EDAD, QUE SEA DE CIERTO PAIS Y TAMBIEN VER EL ESTADO DE SU CONSULTA

```
CREATE FUNCTION estado(@edad int, @pais varchar(20), @estado varchar(10) )
returns varchar(100) as
begin
    declare @respuesta varchar(100)
    select @respuesta = concat(cli.[edad], cli.[pais], con.[decenlace])
    from cliente as cli
    INNER JOIN consultas as con on con.[id cliente] = cli.[id cliente]
    where cli.pais = @pais
    and cli.edad > @edad
    and con.[decenlace] = @estado;
    return @respuesta;
end;

select dbo.estado(15, 'Argentina', 'compra') as DBA;
```

```

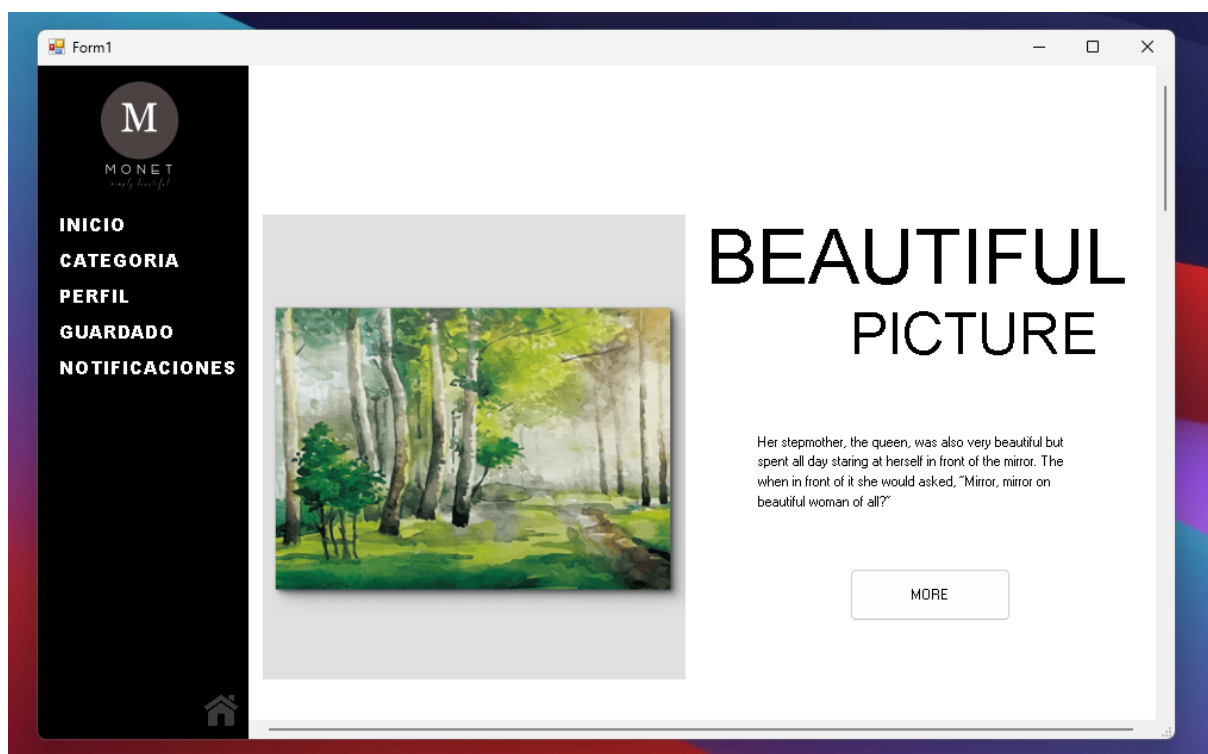
-- crar una funcion que sume todos los id de los clietes
CREATE FUNCTION sumar_elementos()
returns int as
begin
    declare @respuesta int
    select @respuesta = sum([id_cliente])
    from cliente
    return @respuesta;
end;

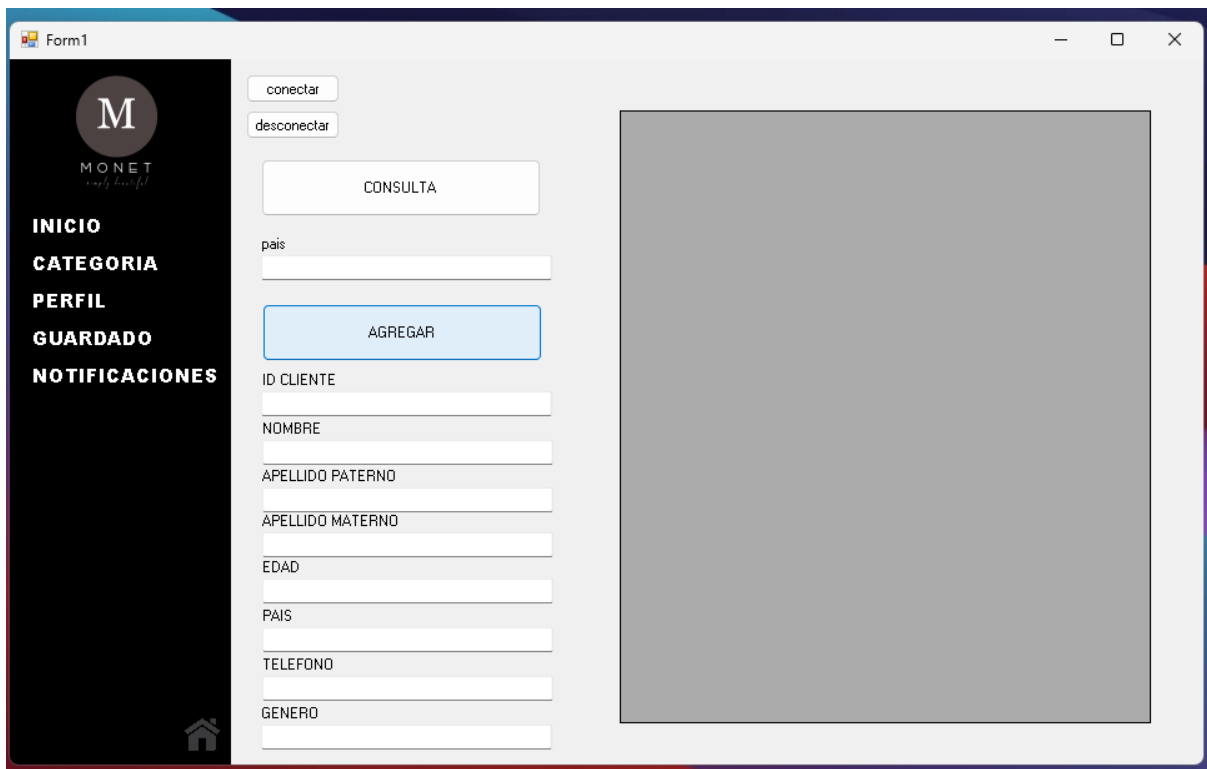
select dbo.sumar_elementos() as DBA;

--funcion que permita contar todos los clientes que sean de nacionalidad Mexicana
CREATE FUNCTION nacionalidad()
returns int as
begin
    declare @res int
    select @res = sum([nombres])
    from cliente AS cli
    where cli.[pais] = 'Mexico'
    return @res;
end;

select dbo.nacionalidad() as DBA;

```





The screenshot shows a web application window titled 'Form1'. On the left is a dark sidebar with a logo 'M MONET' and navigation links: INICIO, CATEGORIA, PERFIL, GUARDADO, and NOTIFICACIONES. The main area contains a form with fields for 'pais', 'ID CLIENTE', 'NOMBRE', 'APELLIDO PATERNO', 'APELLIDO MATERNO', 'EDAD', 'PAIS', 'TELEFONO', and 'GENERO'. There are buttons for 'conectar', 'desconectar', 'CONSULTA', and 'AGREGAR'. To the right of the form is a table with 4 columns: 'id cliente', 'nombres', 'apellido paterno', and 'apellido materno'. The table contains 20 rows of client data.

id cliente	nombres	apellido paterno	apellido materno
123	Jose	Barrera	Fuentes
124	Raul	Torrez	Cruz
125	Regina	Gonzales	Gutierrez
126	Carlos	Medrano	Perez
127	Miguel	Esequiel	Toledo
128	Regina	Vargas	Fuentes
129	Guina	De La Cruz	Paredes
130	Oscar	Toledo	Pinto
131	Fernando	Reyes	Firmino
132	Juan	Gonzales	Garcia
133	Hector	Quispe	Mamani
134	Maria	Lopez	Gutierrez
135	Gabo	Garmendia	Sain
136	Reyna	Torres	Veltran
137	Miguel	Toledo	Medrano
138	Arturo	Herrera	Guygua
139	Josefina	Hernandez	Cruz
140	Patricia	Hernandez	Casablanca
141	Jose	De la Cruz	Barrera
142	Rene	Garcia	Martines

[ENLACE AL VIDEO DE EXPLICACIÓN DEL PROYECTO\(clik\)](#)

CONCLUSIONES

Se puede apreciar que el programa y la base de datos interactúan de manera normal sin ningún inconveniente lo que sí se debería aumentar es la interacción más íntima entre estas pero de cualquier solamente es adjuntas más formularios para cada tabla.

También este proyecto se lo puede desarrollar de manera mucho más profesional para así crear su total funcionalidad y sacarla a las tiendas de aplicaciones por que en cuanto a la base este ya esta realizado solamente es cuestión de mejorarla.

Dado los resultados obtenidos se puede decir que el proyecto marchó como se esperaba, todo está funcionando de manera correcta y no se podía esperar más, ya solo queda en posteriores como se mencionó anteriormente mejorarla para tener una aplicación totalmente funcional para usuario ya reales.