<?php

namespace App\Entidades;

use Illuminate\Database\Eloquent\Model;

use DB;

use Session;

require app\_path().'/start/constants.php';

class Cliente extends Model

{

    protected $table = 'clientes';

    public $timestamps = false;

    protected $fillable = [

        'idcliente', 'nombre', 'apellido', 'dni', 'telefono', 'direccion', 'codigo\_postal', 'fecha\_nac', 'correo', 'fk\_idusuario'

    ];

    protected $hidden = [

    ];

    public function cargarDesdeRequest($request)

    {

        $this->idcliente = $request->input('id') != "0" ? $request->input('id') : $this->idcliente;

        $this->nombre = $request->input('txtNombre');

        $this->apellido = $request->input('txtApellido');

        $this->dni = $request->input('txtDni');

        $this->telefono = $request->input('txtTelefono');

        $this->direccion = $request->input('txtDireccion');

        $this->codigo\_postal = $request->input('txtCodigo\_postal');

        $this->fecha\_nac = $request->input('txtFechaNac');

        $this->correo = $request->input('txtCorreo');

        $this->fk\_idusuario = $request->input('lstUsuario');

    }

    public function obtenerFiltrado()

    {

        $request = $\_REQUEST;

        $columns = array(

            0 => 'A.nombre',

            1 => 'A.apellido',

            2 => 'A.dni',

            3 => 'A.telefono',

            4 => 'A.direccion',

            5 => 'A.codigo\_postal',

            6 => 'A.fecha\_nac',

            7 => 'A.correo',

            8 => 'A.fk\_idusuario'

        );

        $sql = "SELECT DISTINCT

                    A.idcliente,

                    A.nombre,

                    A.apellido,

                    A.dni,

                    A.telefono,

                    A.direccion,

                    A.codigo\_postal,

                    A.fecha\_nac,

                    A.correo,

                    A.fk\_idusuario

                    FROM clientes A

                WHERE 1=1

                ";

        //Realiza el filtrado

        if (!empty($request['search']['value'])) {

            $sql .= " AND ( A.nombre LIKE '%" . $request['search']['value'] . "%'";

            $sql .= " OR A.nombre LIKE '%" . $request['search']['value'] . "%'";

            $sql .= " OR A.apellido LIKE '%" . $request['search']['value'] . "%'";

            $sql .= " OR A.dni LIKE '%" . $request['search']['value'] . "%')";

            $sql .= " OR A.telefono LIKE '%" . $request['search']['value'] . "%')";

            $sql .= " OR A.direccion LIKE '%" . $request['search']['value'] . "%')";

            $sql .= " OR A.codigo\_postal LIKE '%" . $request['search']['value'] . "%')";

            $sql .= " OR A.fecha\_nac LIKE '%" . $request['search']['value'] . "%')";

            $sql .= " OR A.correo LIKE '%" . $request['search']['value'] . "%')";

            $sql .= " OR A.fk\_idusuario LIKE '%" . $request['search']['value'] . "%')";

        }

        $sql .= " ORDER BY " . $columns[$request['order'][0]['column']] . "   " . $request['order'][0]['dir'];

        $lstRetorno = DB::select($sql);

        return $lstRetorno;

    }

    public function obtenerTodos()

    {

        $sql = "SELECT \* FROM clientes ORDER BY nombre";

        $lstRetorno = DB::select($sql);

        return $lstRetorno;

    }

    public function obtenerPorId($idcliente)

    {

        $sql = "SELECT \* FROM clientes WHERE idcliente = $idcliente";

        $lstRetorno = DB::select($sql);

        if (count($lstRetorno) > 0) {

            $this->idcliente = $lstRetorno[0]->idcliente;

            $this->nombre = $lstRetorno[0]->nombre;

            $this->apellido = $lstRetorno[0]->apellido;

            $this->dni = $lstRetorno[0]->dni;

            $this->telefono = $lstRetorno[0]->telefono;

            $this->direccion = $lstRetorno[0]->direccion;

            $this->codigo\_postal = $lstRetorno[0]->codigo\_postal;

            $this->fecha\_nac = $lstRetorno[0]->fecha\_nac;

            $this->correo = $lstRetorno[0]->correo;

            $this->fk\_idusuario = $lstRetorno[0]->fk\_idusuario;

            return $this;

        }

        return null;

    }

   public function obtenerPorUsuario($id){

        $sql ="SELECT

            A.nombre,

            A.apellido

            A.dni,

            A.telefono,

            A.direccion,

            A.codigo\_postal,

            A.fecha\_nac,

            A.correo,

            A.fk\_idusuario,

            B.usuario

            FROM clientes

            LEFT JOIN usuarios B ON A.fk\_idcliente = B.idusuario

            WHERE fk\_idusuario = '$id'";

        $lstRetorno = $sql;

        return $lstRetorno;

   }

    public function guardar()

    {

        $sql = "UPDATE clientes SET

            idcliente = '$this->idcliente',

            nombre = '$this->nombre',

            apellido = '$this->apellido',

            dni = '$this->dni',

            telefono = '$this->telefono',

            direccion = '$this->direccion',

            codigo\_postal = '$this->codigo\_postal',

            fecha\_nac = '$this->fecha\_nac',

            correo = '$this->correo',

            /\*fk\_idpoliza = '$this->fk\_idpoliza',\*/

            fk\_idlocalidad = '$this->fk\_idlocalidad',

            fk\_idusuario = '$this->fk\_idusuario'

            WHERE idcliente=?";

        $affected = DB::update($sql, [$this->idcliente]);

    }

    public function eliminar()

    {

        $sql = "DELETE FROM clientes WHERE

            idcliente=?";

        $affected = DB::delete($sql, [$this->idcliente]);

    }

    public function insertar()

    {

        $sql = "INSERT INTO clientes (

                nombre,

                apellido,

                dni,

                telefono,

                direccion,

                codigo\_postal,

                fecha\_nac,

                correo,

                fk\_idusuario

            ) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?);";

        $result = DB::insert($sql, [

            $this->nombre,

            $this->apellido,

            $this->dni,

            $this->telefono,

            $this->direccion,

            $this->codigo\_postal,

            $this->fecha\_nac,

            $this->correo,

            $this->fk\_idusuario

        ]);

        return $this->idcliente = DB::getPdo()->lastInsertId();

    }

}