Código

1. Perl:

```perl
#!perl/bin/perl.exe

# Recibe: card_id
# Retorna:
# <status>
#     <movement>
#         <amount>cantidad</amount>
#         <type>tipo</type>
#         <date>fecha</date>
#     </movement>
#     <movement>
#         ...
#     </movement>
#     <balance>saldo</balance>
# </status>

use strict;
use warnings;
use CGI;
use CGI::Session;
use CGI::Cookie;
use DBI;

my %type_nums = (d => 1, w => 2, i => 3);
my %type_names = (1 => "DEPOSITO", 2 => "RETIRO", 3 => "TRANSFERENCIA ENVIADA", 4 => "TRANSFERENCIA RECIBIDA");

my $cgi = CGI->new;
$cgi->charset("UTF-8");

my %cookies = CGI::Cookie->fetch();
my $session_cookie = $cookies{"session_id_cliente"};

if ($session_cookie) {
    my $session_id = $session_cookie->value();
    my $session = CGI::Session->load($session_id);
    my $card_id = 1;

    my $u = "query";
    my $p = "YR4AFJUC3nyRmasY";
    my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
    my $dbh = DBI->connect($dsn, $u, $p);

    my $sth = $dbh->prepare("SELECT ‘tarjetas‘.‘moneda‘, ‘movimientos‘.‘monto‘, ‘movimientos‘.‘tipo‘,
        ‘movimientos‘.‘fecha‘
                         FROM tarjetas, movimientos
                         WHERE tarjetas.id = ’$card_id’");
    $sth->execute;

    my @row = $sth->fetchrow_array;
    my $currency = $row[0] eq "s" ? "S/." : "\$";
    my $amount = $currency.$row[1];
    my $type = $type_names{$row[2]};
    my $date = $row[3];
    my $total = $row[1] * get_mult($row[2]);

    print $cgi->header("text/xml");
    print "<status>\n";
    print_row($amount, $type, $date);
    while (@row = $sth->fetchrow_array) {
        $total = $total + ($row[1] * get_mult($row[2]));
        my $amount = $currency.$row[1];
        my $type = $type_names{$row[2]};
        my $date = $row[3];
        print_row($amount, $type, $date);
    }
    my $balance = $currency.$total;
    print<<XML;
        <balance>$balance</balance>
    </status>
XML
}

sub get_mult {
    my $mult = 1;
    if ($type_nums{$_[0]} == 2 || $type_nums{$_[0]} == 3) {
        $mult = -1;
    }
    return $mult;
}

sub print_row {
    my $amount = $_[0];
    my $type = $_[1];
    my $date = $_[2];
    print<<XML;
        <movement>
            <amount>$amount</amount>
            <type>$type</type>
            <date>$date</date>
        </movement>
XML
}
```

```
#!perl/bin/perl.exe

# Recibe: nada
# Retorna:
# <cards>
#     <card>
#         <id>id</id>
#         <card_number>numero de tarjeta</card_number>
#         <code>codigo</code>
#         <expire_date>fecha de expiracion</expire_date>
#         <currency>moneda</currency>
#     </card>
#     <card>
#         ...
#     </card>
# </cards>

use strict;
use warnings;
use CGI;
use CGI::Session;
use CGI::Cookie;
use DBI;

my $cgi = CGI->new;
$cgi->charset("UTF-8");

my %cookies = CGI::Cookie->fetch();
my $session_cookie = $cookies{"session_id_cliente"};

if ($session_cookie) {
    my $session_id = $session_cookie->value();
    my $session = CGI::Session->load($session_id);
    my $account_id = $session->param("account_id");

    my $u = "query";
    my $p = "YR4AFJUC3nyRmasY";
    my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
    my $dbh = DBI->connect($dsn, $u, $p);

    my $sth = $dbh->prepare("SELECT ‘id‘, ‘numero‘, ‘clave‘, ‘vencimiento‘, ‘moneda‘
                         FROM tarjetas
                         WHERE cuenta_id = ’$account_id’");
    $sth->execute;

    while (my @row = $sth->fetchrow_array) {
        my $id = $row[0];
        my $card_number = $row[1];
        my $code = $row[2];
        my $expire_date = $row[3];
        my $currency = $row[4] eq "s" ? "SOLES" : "DÓLARES";

        print $cgi->header("text/xml");
        print "<errors>\n";
        print_card($id, $card_number, $code, $expire_date, $currency);
        print "</errors>\n";
    }
}

sub print_card {
    print<<XML;
        <card>
            <id>$_[0]</id>
            <card_number>$_[1]</card_number>
            <code>$_[2]</code>
            <expire_date>$_[3]</expire_date>
            <currency>$_[4]</currency>
        </card>
XML
}
```

```
#!perl/bin/perl.exe

# Recibe: user, password, type (cliente u operario)
# Retorna:
# <errors>
#    <error>
#        <element>elemento</element>
#        <message>mensaje</message>
#    </error>
#    <error>
#        ...
#    </error>
# </errors>
# o solo la creacion de una cookie si todo esta bien :>

use strict;
use warnings;
use CGI;
use CGI::Session;
use CGI::Cookie;
use DBI;

my $cgi = CGI->new;
$cgi->charset("UTF-8");
my $user = $cgi->param("user");
my $password = $cgi->param("password");
my $type = $cgi->param("type");
my $session_time = 600;

my $user_status = check_user($user);
my $password_status = check_password($password);
my $type_status = check_type($type);
my %errors = (user => $user_status, password => $password_status, type => $type_status);

login();

sub login {
    if (!$user_status && !$password_status && !$type_status) {
        my $u = "query";
        my $p = "YR4AFJUC3nyRmasY";
        my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
        my $dbh = DBI->connect($dsn, $u, $p);

        my $query;
        if ($type eq "cliente") {
            $query = "SELECT ‘cuentas‘.‘id‘, ‘clientes‘.‘id‘, CONCAT(‘clientes‘.‘nombres‘, ’ ’, ‘clientes‘.‘paterno‘,
                 ’ ’, ‘clientes‘.‘materno‘)
                    FROM cuentas, clientes
                    WHERE cuentas.usuario = ’$user’ AND cuentas.clave = ’$password’ AND cuentas.cliente_id =
                         clientes.id";
        } else {
            $query = "SELECT ‘nombre‘ FROM operarios WHERE usuario=’$user’ AND clave=’$password’";
        }
        my $sth = $dbh->prepare($query);
        $sth->execute;

        my @person = $sth->fetchrow_array;
        if (@person) {
            my $session = CGI::Session->new();
            if ($type eq "cliente") {
                $session->param("account_id", $person[0]);
                $session->param("client_id", $person[1]);
                $session->param("name", $person[2]);
            } else {
                $session->param("name", $person[0]);
            }
            $session->expire(time + $session_time);
            $session->flush();

            my $cookie = $cgi->cookie(-name => "session_id_$type",
                                    -value => $session->id(),
                                    -expires => time + $session_time,
                                    "-max-age" => time + $session_time);
            $cookie->bake;

            return;
        } else {
            $errors{login} = "El usuario y la clave no coinciden.";
        }
    }

    print_errors();
}

```

```
sub check_user {
    my $card_number = $_[0];
    my $length = length($card_number);
    if (!$card_number || $length == 0) {
        return "Ingrese un usuario.";
    }
    if ($length > 30) {
        return "Usuario muy largo.";
    }
}

sub check_password {
    my $password = $_[0];
    my $length = length($password);
    if (!$password || $length == 0) {
        return "Ingrese una clave.";
    }
    if ($length > 30) {
        return "Clave muy larga.";
    }
}

sub check_type {
    my $type = $_[0];
    if ($type ne "cliente" && $type ne "operario") {
        return "Tipo no valido.";
    }
}

sub print_errors {
    print $cgi->header("text/xml");
    print "<errors>\n";
    for my $key (keys %errors) {
        if ($errors{$key}) {
        print<<XML;
    <error>
        <element>$key</element>
        <message>$errors{$key}</message>
    </error>
XML
        }
    }
    print "</errors>\n";
}
```

```
#!perl/bin/perl.exe

# Recibe: card_id, type (d, w, i), amount, other_card (only for i)
# deposit, withdrawal, transferin
# Retorna:
# <errors>
#     <error>
#         <element>elemento</element>
#         <message>mensaje</message>
#     </error>
#     <error>
#         ...
#     </error>
# </errors>
# o nada si todo esta bien :>

use strict;
use warnings;
```

```perl
use CGI;
use CGI::Session;
use CGI::Cookie;
use DBI;

my %type_nums = (d => 1, w => 2, i => 3);

my $cgi = CGI->new;
$cgi->charset("UTF-8");
my $card_id = "032";
my $type = $cgi->param("type");
my $amount = $cgi->param("amount");

my %cookies = CGI::Cookie->fetch();
my $session_cookie = $cookies{"session_id_cliente"};

my $u = "query";
my $p = "YR4AFJUC3nyRmasY";
my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
my $dbh = DBI->connect($dsn, $u, $p);

my $card_id_status = check_card_id($card_id);
my $type_status = check_type($type);
my $amount_status = check_amount($amount);

my %errors = (card_id => $card_id_status, type => $type_status, amount => $amount_status);

if ($session_cookie) {
   my $session_id = $session_cookie->value();
   my $session = CGI::Session->load($session_id);
   my $account_id = $session->param("account_id");

   transaction();
}

sub transaction {
   if (!$card_id_status && !$amount_status && !$type_status) {
      my $type_num = $type_nums{$type};
      if ($type_num == 3) {
         my $other_card = $cgi->param("other_card");
         my $sth = $dbh->prepare("SELECT ‘id‘ FROM tarjetas WHERE numero = ’$other_card’");
         $sth->execute;
         my @row = $sth->fetchrow_array;
         my $other_id = $row[0];

         $sth = $dbh->prepare("INSERT INTO movimientos (tarjeta_id, monto, tipo) VALUES ($other_id, $amount, 4)");
         $sth->execute;
      }
      my $sth = $dbh->prepare("INSERT INTO movimientos (tarjeta_id, monto, tipo) VALUES ($card_id, $amount, $type_num)");
      $sth->execute;
      print $cgi->header("text/xml");

      return;
   }

   print_errors();
}

sub check_card_id {
   my $card_id = $_[0];
   my $length = length($card_id);
   if (!$card_id || $length == 0) {
      return "Tarjeta no valida.";
```

```
    }
}

sub check_amount {
    my $amount = $_[0];
    if (!$amount) {
        return "Ingrese una cantidad.";
    }
    if ($amount !~ /^\d+$/ || $amount < 0) {
        return "Cantidad no valida.";
    }
    if ($type eq "w") {
        my $sth = $dbh->prepare("SELECT SUM(monto*tipo) FROM movimientos WHERE tarjeta_id = $card_id");
        $sth->execute;
        my @sum = $sth->fetchrow_array;
        my $balance = $sum[0] || 0;
        if ($balance - $amount < 0) {
            return "Balance negativo."
        }
    }
}

sub check_type {
    my $type = $_[0];
    if (!$type) {
        return "Marque un tipo";
    }
    if ($type !~ /^[dwi]$/) {
        return "Tipo no valido";
    }
}

sub print_errors {
    print $cgi->header("text/xml");
    print "<errors>\n";
    for my $key (keys %errors) {
        if ($errors{$key}) {
        print<<XML;
    <error>
        <element>$key</element>
        <message>$errors{$key}</message>
    </error>
XML
        }
    }
    print "</errors>\n";
}
```

```
#!perl/bin/perl.exe

# Recibe: dni, name, plastname, mlastname, bdate
# Retorna:
# <errors>
#    <error>
#        <element>elemento</element>
#        <message>mensaje</message>
#    </error>
#    <error>
#        ...
#    </error>
# </errors>
# o nada si todo esta bien :>

```

```perl
use strict;
use warnings;
use CGI;
use CGI::Session;
use CGI::Cookie;
use DBI;

my $cgi = CGI->new;
$cgi->charset("UTF-8");
my $dni = $cgi->param("dni");
my $name = $cgi->param("name");
my $plastname = $cgi->param("plastname");
my $mlastname = $cgi->param("mlastname");
my $bdate = $cgi->param("bdate");

my $dni_status = check_DNI($dni);
my $name_status = check_name($name);
my $plastname_status = check_plastname($plastname);
my $mlastname_status = check_mlastname($mlastname);
my $bdate_status = check_bdate($bdate);

my %errors = (dni => $dni_status, name => $name_status, plastname => $plastname_status, mlastname => $mlastname_status, bdate => $bdate_status);

register();

sub register {
    if (!$dni_status && !$name_status && !$plastname_status && !$mlastname_status && !$bdate_status) {
        my $u = "query";
        my $p = "YR4AFJUC3nyRmasY";
        my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
        my $dbh = DBI->connect($dsn, $u, $p);

        my $sth = $dbh->prepare("INSERT INTO clientes (dni, nombres, paterno, materno, nacimiento) VALUES ($dni, $name, $plastname, $mlastname, $bdate)");
        $sth->execute();
        print $cgi->header("text/xml");

        return;
    }

    print_errors();
}

sub check_DNI {
    my $dni = $_[0];
    if (!$dni) {
        return "Ingrese un DNI.";
    }
    if ($dni !~ /^\d{8}$/) {
        return "DNI no valido.";
    }
}

sub check_name {
    my $name = $_[0];
    if (!$name) {
        return "Ingrese un nombre.";
    }
    if (length($name) > 40) {
        return "Nombre muy largo.";
    }
    if ($name !~ /^\w{1,40}$/) {
        return "Nombre no valido.";
    }
}

sub check_plastname {
    my $plastname = $_[0];
    my $length = length($plastname);
    if (!$plastname) {
        return "Ingrese un apellido paterno.";
    }
    if ($length > 30) {
        return "Apellido paterno muy largo.";
    }
    if ($plastname !~ /^\w{1,30}$/) {
        return "Apellido paterno no valido.";
    }
}

sub check_mlastname {
    my $mlastname = $_[0];
    my $length = length($mlastname);
    if (!$mlastname) {
        return "Ingrese un apellido materno.";
    }
    if ($length > 30) {
        return "Apellido materno muy largo.";
    }
    if ($mlastname !~ /^\w{1,30}$/) {
        return "Apellido materno no valido.";
    }
}

sub check_bdate {
    my $bdate = $_[0];
    if (!$bdate) {
        return "Ingrese una fecha de nacimiento.";
    }
    if ($bdate !~ /^\d{4}\-\d{2}\-\d{2}$/) {
        return "Fecha de nacimiento no valida.";
    }
}

sub print_errors {
    print $cgi->header("text/xml");
    print "<errors>\n";
    for my $key (keys %errors) {
        if ($errors{$key}) {
        print<<XML;
    <error>
        <element>$key</element>
        <message>$errors{$key}</message>
    </error>
XML
        }
    }
    print "</errors>\n";
}
```

```perl
#!perl/bin/perl.exe

# Recibe: user, password, dni
# Retorna:
# <errors>
#     <error>
#        <element>elemento</element>
#        <message>mensaje</message>
#    </error>
#    <error>
#        ...
#    </error>
# </errors>
# o nada si todo esta bien :>

use strict;
use warnings;
use CGI;
use CGI::Session;
use CGI::Cookie;
use DBI;

my $cgi = CGI->new;
$cgi->charset("UTF-8");
my $user = $cgi->param("user");
my $password = $cgi->param("password");
my $dni = $cgi->param("dni");

our $client_id;
my $user_status = check_user($user);
my $password_status = check_password($password);
my $dni_status = check_DNI($dni);

my %errors = (user => $user_status, password => $password_status, dni => $dni_status);

register();

sub register {
    if (!$user_status && !$password_status && !$dni_status) {
        my $u = "query";
        my $p = "YR4AFJUC3nyRmasY";
        my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
        my $dbh = DBI->connect($dsn, $u, $p);

        my $sth = $dbh->prepare("INSERT INTO cuentas (usuario, clave, cliente_id) VALUES ($user, $password, $client_id)");
        $sth->execute;
        print $cgi->header("text/xml");

        return;
    }

    print_errors();
}

sub check_user {
    my $card_number = $_[0];
    my $length = length($card_number);
    if (!$card_number || $length == 0) {
        return "Ingrese un usuario.";
    }
    if ($length > 30) {
        return "Usuario muy largo.";
    }
}

sub check_password {
    my $password = $_[0];
    my $length = length($password);
    if (!$password || $length == 0) {
        return "Ingrese una clave.";
    }
    if ($length > 30) {
        return "Clave muy larga.";
    }
}

sub check_DNI {
    my $dni = $_[0];
    if (!$dni) {
        return "Ingrese un DNI.";
    }
    if ($dni !˜ /ˆ\d{8}$/) {
        return "DNI no valido.";
    }
    my $sth = $dbh->prepare("SELECT ‘id‘ FROM clientes WHERE dni = $dni");
    $sth->execute;
    my @row = $sth->fetchrow_array;
    if (!@row) {
        return "Cliente no existente.";
    }
    $client_id = $row[0];
}

sub print_errors {
    print $cgi->header("text/xml");
    print "<errors>\n";
    for my $key (keys %errors) {
        if ($errors{$key}) {
        print<<XML;
    <error>
        <element>$key</element>
        <message>$errors{$key}</message>
    </error>
XML
        }
    }
    print "</errors>\n";
}
```

```
#!perl/bin/perl.exe

# Recibe: number, password, currency (s o d), dni
# Retorna:
# <errors>
#     <error>
#         <element>elemento</element>
#         <message>mensaje</message>
#     </error>
#     <error>
#         ...
#     </error>
# </errors>
# o nada si todo esta bien :>

use strict;
use warnings;
use CGI;
use CGI::Session;
use CGI::Cookie;
use DBI;
use DateTime;

```

```perl
my $cgi = CGI->new;
$cgi->charset("UTF-8");
my $number = $cgi->param("number");
my $password = $cgi->param("password");
my $currency = $cgi->param("currency");
my $dni = $cgi->param("dni");
my $current_date = DateTime->now;
my $expire_date = $current_date->add(years => 7)->ymd;

our $account_id;
my $number_status = check_number($number);
my $password_status = check_password($password);
my $currency_status = check_currency($currency);
my $dni_status = check_DNI($dni);

register();

sub register {
    if (!$number_status && !$password_status && !$currency_status && !$dni_status) {
        my $u = "query";
        my $p = "YR4AFJUC3nyRmasY";
        my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
        my $dbh = DBI->connect($dsn, $u, $p);

        my $sth = $dbh->prepare("INSERT INTO tarjetas (numero, clave, vencimiento, cuenta_id) VALUES (?, ?, ?)");
        $sth->execute($number, $password, $expire_date);
        print $cgi->header("text/xml");

        return;
    }

    print_errors();
}

sub check_number {
    my $number = $_[0];
    if (!$number) {
        return "Ingrese un número de tarjeta.";
    }
    if ($number !~ /^\d{16}$/) {
        return "Número de tarjeta no valido.";
    }
}

sub check_password {
    my $password = $_[0];
    if (!$password) {
        return "Ingrese una clave.";
    }
    if (length($password) > 30) {
        return "Clave muy larga.";
    }
    if (length($password) < 8) {
        return "Clave muy corta.";
    }
}

sub check_currency {
    my $currency = $_[0];
    if (!$currency) {
        return "Marque una moneda.";
    }
    if ($currency !~ /^[sd]$/) {
        return "Moneda no valida.";
    }
}

sub check_DNI {
    my $dni = $_[0];
    if (!$dni) {
        return "Ingrese un DNI.";
    }
    if ($dni !~ /^\d{8}$/) {
        return "DNI no valido.";
    }
    my $sth = $dbh->prepare("SELECT ‘cuenta‘.‘id‘
                         FROM cuentas, clientes
                         WHERE clientes.dni = ’$dni’ AND cuentas.cliente_id = ’clientes_id’");
    $sth->execute;
    my @row = $sth->fetchrow_array;
    if (!@row) {
        return "Cliente no existente."
    }
    $client_id = $row[0];
}

sub print_errors {
    print $cgi->header("text/xml");
    print "<errors>\n";
    for my $key (keys %errors) {
        if ($errors{$key}) {
        print<<XML;
    <error>
        <element>$key</element>
        <message>$errors{$key}</message>
    </error>
XML
        }
    }
    print "</errors>\n";
}
```

```
#!perl/bin/perl.exe

# Recibe: action (check [revisa si la sesion esta activa], close [cierra la sesion]), type (cliente u operario)
# Retorna:
# <session>
#     <logged_in>0 o 1</logged_in>
#     <name>nombre</name>
#     <expire_time>tiempo restante</expire_time>
# </session>

use strict;
use warnings;
use CGI;
use CGI::Session;
use CGI::Cookie;

my $cgi = CGI->new;
$cgi->charset("UTF-8");
my $action = $cgi->param("action");
my $type = $cgi->param("type");

my %cookies = CGI::Cookie->fetch();
my $session_cookie = $cookies{"session_id_$type"};

print ($cgi->header("text/xml"));
```

```perl
print "<session>\n";
if ($action eq "check") {
    if ($session_cookie) {
        my $session_id = $session_cookie->value();
        my $session = CGI::Session->load($session_id);
        my $name = $session->param("name");
        my $expire_time = $session->expires() - time;
        print<<XML;
        <logged_in>1</logged_in>
        <name>$name</name>
        <expire_time>$expire_time</expire_time>
XML
    } else {
        print "<logged_in>0</logged_in>\n";
    }
} elsif ($action eq "close") {
    print "<logged_in>0</logged_in>\n";
    if ($session_cookie) {
        my $session_id = $session_cookie->value();
        my $session = CGI::Session->load($session_id);
        $session_cookie->expires("-1h");
        $session->delete();
        $session->flush();
    }
}
print "</session>";
```

2. SQL:

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Servidor: 127.0.0.1
-- Tiempo de generación: 17-01-2024 a las 04:43:38
-- Versión del servidor: 10.4.28-MariaDB
-- Versión de PHP: 8.2.4

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de datos: ‘bancafinal‘
--

-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla ‘clientes‘
--

CREATE TABLE ‘clientes‘ (
  ‘id‘ int(11) NOT NULL,
  ‘dni‘ varchar(8) NOT NULL,
  ‘nombres‘ varchar(40) NOT NULL,
  ‘paterno‘ varchar(30) NOT NULL,
  ‘materno‘ varchar(30) NOT NULL,
  ‘nacimiento‘ date NOT NULL,
  ‘creado‘ timestamp NOT NULL DEFAULT current_timestamp(),
  ‘estado‘ tinyint(1) NOT NULL DEFAULT 1
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla ‘clientes‘
--

INSERT INTO ‘clientes‘ (‘id‘, ‘dni‘, ‘nombres‘, ‘paterno‘, ‘materno‘, ‘nacimiento‘, ‘creado‘, ‘estado‘) VALUES
(1, ’54587912’, ’Christian Raul’, ’Mestas’, ’Zegarra’, ’2006-04-17’, ’2024-01-17 02:19:50’, 1);

-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla ‘cuentas‘
--

CREATE TABLE ‘cuentas‘ (
  ‘id‘ int(11) NOT NULL,
  ‘usuario‘ varchar(30) NOT NULL,
  ‘clave‘ varchar(30) NOT NULL,
  ‘creado‘ timestamp NOT NULL DEFAULT current_timestamp(),
  ‘estado‘ tinyint(1) NOT NULL DEFAULT 1,
  ‘cliente_id‘ int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
--
-- Volcado de datos para la tabla ‘cuentas‘
--

INSERT INTO ‘cuentas‘ (‘id‘, ‘usuario‘, ‘clave‘, ‘creado‘, ‘estado‘, ‘cliente_id‘) VALUES
(1, ’cmestasz’, ’123456789’, ’2024-01-17 02:20:14’, 1, 1);

-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla ‘movimientos‘
--

CREATE TABLE ‘movimientos‘ (
  ‘id‘ int(11) NOT NULL,
  ‘tarjeta_id‘ int(11) NOT NULL,
  ‘monto‘ float NOT NULL,
  ‘tipo‘ tinyint(4) NOT NULL,
  ‘fecha‘ timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla ‘movimientos‘
--

INSERT INTO ‘movimientos‘ (‘id‘, ‘tarjeta_id‘, ‘monto‘, ‘tipo‘, ‘fecha‘) VALUES
(3, 1, 50000, 1, ’2024-01-17 02:21:50’),
(5, 1, 25000, -1, ’2024-01-17 02:23:23’);

-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla ‘operarios‘
--

CREATE TABLE ‘operarios‘ (
  ‘id‘ int(11) NOT NULL,
  ‘nombre‘ varchar(30) NOT NULL,
  ‘usuario‘ varchar(30) NOT NULL,
  ‘clave‘ varchar(30) NOT NULL,
  ‘creado‘ timestamp NOT NULL DEFAULT current_timestamp(),
  ‘estado‘ tinyint(1) NOT NULL DEFAULT 1
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla ‘tarjetas‘
--

CREATE TABLE ‘tarjetas‘ (
  ‘id‘ int(11) NOT NULL,
  ‘numero‘ varchar(16) NOT NULL,
  ‘clave‘ varchar(30) NOT NULL,
  ‘creado‘ timestamp NOT NULL DEFAULT current_timestamp(),
  ‘vencimiento‘ datetime NOT NULL,
  ‘sellado‘ tinyint(1) NOT NULL DEFAULT 1,
  ‘estado‘ tinyint(1) NOT NULL DEFAULT 1,
  ‘moneda‘ varchar(1) NOT NULL,
  ‘cuenta_id‘ int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
```

```sql
-- Volcado de datos para la tabla ‘tarjetas‘
--

INSERT INTO ‘tarjetas‘ (‘id‘, ‘numero‘, ‘clave‘, ‘creado‘, ‘vencimiento‘, ‘sellado‘, ‘estado‘, ‘moneda‘, ‘cuenta_id‘) VALUES
(1, ’5565894512347787’, ’123456789’, ’2024-01-17 02:20:32’, ’2024-01-17 03:20:19’, 1, 1, ’s’, 1);

--
-- Índices para tablas volcadas
--

--
-- Indices de la tabla ‘clientes‘
--
ALTER TABLE ‘clientes‘
  ADD PRIMARY KEY (‘id‘),
  ADD UNIQUE KEY ‘dni‘ (‘dni‘);

--
-- Indices de la tabla ‘cuentas‘
--
ALTER TABLE ‘cuentas‘
  ADD PRIMARY KEY (‘id‘),
  ADD KEY ‘cliente_id‘ (‘cliente_id‘) USING BTREE;

--
-- Indices de la tabla ‘movimientos‘
--
ALTER TABLE ‘movimientos‘
  ADD PRIMARY KEY (‘id‘),
  ADD KEY ‘tarjeta_id‘ (‘tarjeta_id‘) USING BTREE;

--
-- Indices de la tabla ‘operarios‘
--
ALTER TABLE ‘operarios‘
  ADD PRIMARY KEY (‘id‘);

--
-- Indices de la tabla ‘tarjetas‘
--
ALTER TABLE ‘tarjetas‘
  ADD PRIMARY KEY (‘id‘),
  ADD UNIQUE KEY ‘numero‘ (‘numero‘),
  ADD KEY ‘cuenta_id‘ (‘cuenta_id‘) USING BTREE;

--
-- AUTO_INCREMENT de las tablas volcadas
--

--
-- AUTO_INCREMENT de la tabla ‘clientes‘
--
ALTER TABLE ‘clientes‘
  MODIFY ‘id‘ int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT de la tabla ‘cuentas‘
--
ALTER TABLE ‘cuentas‘
  MODIFY ‘id‘ int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT de la tabla ‘movimientos‘
```

```
--
ALTER TABLE ‘movimientos‘
  MODIFY ‘id‘ int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT de la tabla ‘operarios‘
--
ALTER TABLE ‘operarios‘
  MODIFY ‘id‘ int(11) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT de la tabla ‘tarjetas‘
--
ALTER TABLE ‘tarjetas‘
  MODIFY ‘id‘ int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- Restricciones para tablas volcadas
--

--
-- Filtros para la tabla ‘cuentas‘
--
ALTER TABLE ‘cuentas‘
  ADD CONSTRAINT ‘cliente_id‘ FOREIGN KEY (‘cliente_id‘) REFERENCES ‘clientes‘ (‘id‘) ON DELETE CASCADE ON UPDATE
      CASCADE;

--
-- Filtros para la tabla ‘movimientos‘
--
ALTER TABLE ‘movimientos‘
  ADD CONSTRAINT ‘movimientos_ibfk_1‘ FOREIGN KEY (‘tarjeta_id‘) REFERENCES ‘tarjetas‘ (‘id‘) ON DELETE CASCADE ON
      UPDATE CASCADE;

--
-- Filtros para la tabla ‘tarjetas‘
--
ALTER TABLE ‘tarjetas‘
  ADD CONSTRAINT ‘tarjetas_ibfk_1‘ FOREIGN KEY (‘cuenta_id‘) REFERENCES ‘cuentas‘ (‘id‘) ON DELETE CASCADE ON UPDATE
      CASCADE;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

-- usuarios
CREATE USER ’query’@’%’ IDENTIFIED BY ’YR4AFJUC3nyRmasY’;
GRANT SELECT, INSERT ON ‘bancafinal‘.* TO ’query’@’%’;
```

3. HTML:

```html
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8" />
    <title>Estado de cuenta</title>
    <link rel="stylesheet" href="./css/estilos_generales.css" />
    <link rel="stylesheet" href="./css/menus_clientes.css" />
    <script src="./scripts/utilidades.js"></script>
    <script src="./scripts/menus_clientes.js"></script>
    <script src="./scripts/estado_tarjeta.js"></script>
</head>

<body>
    <div id="content" class="content">
        <div class="top">
            <h2>Estado de cuenta</h2>
            <div class="rigthTop">
                <a href="./index_clientes.html" class="btnLink">Volver</a>
                <div>
                    <h4 id="timer">Su sesión concluirá en:</h4>
                    <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
                </div>
            </div>
        </div>
        <div class="form">
            <h3>Balance de cuenta:</h3>
            <div id="balance"></div>
            <h3>Últimos movimientos:</h3>
        </div>
        <div id="movements" style="width: 100 %;">
            <div class="movement">
                <h5 style="font-size: 20px;">CANTIDAD</h5>
                <h5 style="font-size: 20px;">TIPO</h5>
                <h5 style="font-size: 20px;">FECHA</h5>
            </div>
        </div>
    </div>
</body>

</html>
```

```html
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8" />
    <title>Usuario</title>
    <link rel="stylesheet" href="./css/estilos_generales.css" />
    <link rel="stylesheet" href="./css/menus_clientes.css" />
    <script src="./scripts/menus_clientes.js"></script>
    <script src="./scripts/utilidades.js"></script>
</head>

<body>
    <div id="content" class="content">
        <div class="top">
            <h2>
                Acceso rápido
                <div id="welcome"></div>
            </h2>
            <div class="rigthTop">
                <a href="./movimiento_cliente.html" class="btnLink">Realizar un movimiento</a>
                <a href="./estado_tarjeta.html" class="btnLink">Ver estado</a>
                <div>
                    <h4 id="timer">Su sesión concluirá en:</h4>
                    <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
                </div>
            </div>
        </div>
    </div>
</body>

</html>
```

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8" />
    <title>Usuario</title>
    <link rel="stylesheet" href="./css/estilos_generales.css" />
    <link rel="stylesheet" href="./css/menus_operarios.css" />
    <script src="./scripts/utilidades.js"></script>
    <script src="./scripts/menus_operarios.js"></script>
</head>

<body>
    <div id="content" class="content">
        <div class="top">
            <h2>
                Acceso rápido
                <div id="welcome"></div>
            </h2>
            <div>
                <h4 id="timer">Su sesión concluirá en:</h4>
                <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
            </div>
        </div>
        <div class="menu">
            <a href="./registro_clientes.html" class="btnLink">Ingresar nuevo cliente</a>
            <a href="./registro_cuentas.html" class="btnLink">Crear nueva cuenta</a>
            <a href="./registro_tarjetas.html" class="btnLink">Crear nueva tarjeta</a>
        </div>
    </div>
</body>

</html>
```

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8" />
    <title>Ingreso de Clientes</title>
    <link rel="stylesheet" href="./css/estilos_generales.css" />
    <link rel="stylesheet" href="./css/login_clientes.css" />
    <script src="./scripts/utilidades.js"></script>
    <script src="./scripts/login_clientes.js"></script>
</head>

<body>
```

```
    <div id="content" class="content">
        <img src="./logo.png" onclick="redirect(’./login_clientes.html’)" class="logo" />
    </div>
    <div id="form" class="form">
        <h1>Ingreso de Clientes</h1>
        <input type="text" id="user" class="inputText" placeholder="Usuario" required />
        <input type="password" id="password" placeholder="Contraseña" class="inputText" />
        <input type="button" class="submitButton" onclick="login()" value="Ingresar"/>
    </div>
</body>

</html>
```

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8" />
    <title>Ingreso de Operarios</title>
    <link rel="stylesheet" href="./css/estilos_generales.css" />
    <link rel="stylesheet" href="./css/login_operarios.css" />
    <script src="./scripts/utilidades.js"></script>
    <script src="./scripts/login_operarios.js"></script>
</head>

<body>
    <div id="content" class="content">
        <img src="./logo.png" onclick="redirect(’./login_operarios.html’)" class="logo" />
        <h1>Ingreso de Operarios</h1>
        <div id="form" class="form">
            <input type="text" id="user" placeholder="Usuario" class="inputText" />
            <input type="password" id="password" placeholder="Contraseña" class="inputText" />
            <input type="button" id="login" class="submitButton" onclick="login()" value="Ingresar" />
        </div>
    </div>
</body>

</html>
```

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8" />
    <title>Realizar un movimiento</title>
    <link rel="stylesheet" href="./css/estilos_generales.css" />
    <link rel="stylesheet" href="./css/menus_clientes.css" />
    <script src="./scripts/utilidades.js"></script>
    <script src="./scripts/menus_clientes.js"></script>
    <script src="./scripts/movimiento_cliente.js"></script>
</head>

<body>
    <div id="content" class="content">
        <div class="top">
            <h2>
                Asistente de movimientos
                <div id="welcome"></div>
            </h2>
            <div class="rigthTop">
                <a href="./index_clientes.html" class="btnLink">Volver</a>
                <div>
```

```
                    <h4 id="timer">Su sesión concluirá en:</h4>
                    <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
                </div>
            </div>
        </div>


        <div id="form" class="form">
            <h3>Tipo de movimiento:</h3>
            <div id="type" class="options">
                <input type="radio" id="deposit" name="type" value="d" checked />
                <label for="d">Depósito</label>
                <input type="radio" id="withdrawal" name="type" value="w" />
                <label for="w">Retiro</label>
                <input type="radio" id="transference" name="type" value="t" />
                <label for="t">Transferencia</label>
            </div>
            <input type="text" id="amount" placeholder="Cantidad" class="inputText" />
            <input type="button" class="submitButton" onclick="doMovement()" value="Realizar movimiento" />
        </div>
    </div>
</body>

</html>
```

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8" />
    <title>Creación de clientes</title>
    <link rel="stylesheet" href="./css/estilos_generales.css" />
    <link rel="stylesheet" href="./css/menus_operarios.css" />
    <script src="./scripts/utilidades.js"></script>
    <script src="./scripts/menus_operarios.js"></script>
    <script src="./scripts/registro_clientes.js"></script>
</head>

<body>
    <div id="content" class="content">
        <div class="top">
            <div id="welcome"></div>
            <div>
                <h4 id="timer">Su sesión concluirá en:</h4>
                <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
            </div>
        </div>
        <div style="height: 15px"></div>
        <a href="./index_operarios.html">Volver</a>
        <div style="height: 10px"></div>
        <h2>Asistente de creación de clientes</h2>
        <div style="height: 25px"></div>
        <input type="text" id="dni" placeholder="DNI" />
        <div style="height: 25px"></div>
        <input type="text" id="name" placeholder="Nombres completos" />
        <div style="height: 25px"></div>
        <input type="text" id="plastname" placeholder="Apellido paterno" />
        <div style="height: 25px"></div>
        <input type="text" id="mlastname" placeholder="Apellido materno" />
        <div style="height: 25px"></div>
        <input type="date" id="bdate" placeholder="Fecha de nacimiento" />
        <div style="height: 25px"></div>
        <input type="button" class="submit" onclick="createClient()" value="Crear cliente" />
    </div>
</body>

</html>
```

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8" />
    <title>Creación de cuentas</title>
    <link rel="stylesheet" href="./css/estilos_generales.css" />
    <link rel="stylesheet" href="./css/menus_operarios.css" />
    <script src="./scripts/utilidades.js"></script>
    <script src="./scripts/menus_operarios.js"></script>
    <script src="./scripts/registro_cuentas.js"></script>
</head>

<body>
    <div id="content" class="content">
        <div class="top">
            <div id="welcome"></div>
            <div>
               <h4 id="timer">Su sesión concluirá en:</h4>
            <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
            </div>
        </div>
        <div style="height: 15px"></div>
        <a href="./index_operarios.html">Volver</a>
        <div style="height: 10px"></div>
        <h2>Asistente de creación de cuentas</h2>
        <div style="height: 25px"></div>
        <input type="text" id="number" placeholder="Número de tarjeta" />
        <div style="height: 25px"></div>
        <input type="button" class="submit" onclick="redirect(’./registro_tarjetas.html’)"
            value="Creación de tarjetas" />
        <div style="height: 25px"></div>
        <h3>Tipo de moneda:</h3>
        <div id="currency" class="options">
            <input type="radio" id="soles" name="currency" value="s" checked />
            <label for="soles">Soles</label>
            <div style="width: 50px"></div>
            <input type="radio" id="dollars" name="currency" value="d" />
            <label for="dollars">Dólares</label>
        </div>
        <div style="height: 45px"></div>
        <input type="text" id="dni" placeholder="DNI del cliente" />
        <div style="height: 25px"></div>
        <input type="button" class="submit" onclick="createAccount()" value="Crear cuenta" />
    </div>
</body>

</html>
```

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8" />
    <title>Creación de tarjetas</title>
    <link rel="stylesheet" href="./css/estilos_generales.css" />
    <link rel="stylesheet" href="./css/menus_operarios.css" />
```

```html
    <script src="./scripts/utilidades.js"></script>
    <script src="./scripts/menus_operarios.js"></script>
    <script src="./scripts/registro_tarjetas.js"></script>
</head>

<body>
    <div id="content" class="content">
        <div class="top">
            <div id="welcome"></div>
            <div>
                <h4 id="timer">Su sesión concluirá en:</h4>
                <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
            </div>
        </div>
        <div style="height: 15px"></div>
        <a href="./index_operarios.html">Volver</a>
        <div style="height: 10px"></div>
        <h2>Asistente de creación de cuentas</h2>
        <div style="height: 25px"></div>
        <input type="button" class="submit" onclick="generateNumber()" value="Generar número" />
        <div style="height: 25px"></div>
        <input type="text" id="number" placeholder="Número de tarjeta" />
        <div style="height: 25px"></div>
        <input type="button" class="submit" onclick="generatePassword()" value="Generar clave" />
        <div style="height: 25px"></div>
        <input type="text" id="password" placeholder="Clave" />
        <div style="height: 25px"></div>
        <input type="button" class="submit" onclick="createCard()" value="Crear tarjeta" />
    </div>
</body>

</html>
```

4. CSS:

```css
/* Body Style */
body {
    font-family: sans-serif;
    background: linear-gradient(to top right, #d9ebfd, #fef9de);
    min-height: 100vh;
    margin: 0;
    padding: 0;
}

/* Content Style */

.top {
    box-sizing: border-box;
    display: flex;
    width: 100%;
    max-width: 1400px;
    justify-content: space-between;
    align-items: center;
    padding: 0px 40px;
}

@media screen and (max-width: 700px) {
    .top {
        flex-direction: column;
    }
}

/* Header Styles */

/* Titles */
h1 {
    text-align: center;
    font-size: 40px;
    color: #002a8d;
}

/* Subtitles */
h2 {
    text-align: left;
    font-size: 35px;
    color: #002a8d;
}

/* Messages */
h3 {
    text-align: center;
    font-size: 20px;
    color: #002a8d;
}

.error {
    color: #e30425;
}

/* Text */
h4 {
    text-align: center;
    font-size: 15px;
    color: #202e44;
}

```

```
/* Text - Inline */
h5 {
    flex: 1;
    display: inline-block;
    text-align: center;
    font-size: 15px;
    color: #202e44;
}

/* Links */
a {
    text-align: left;
    font-size: 20px;
    text-decoration: none;
    color: #202e44;
}

a:hover {
    text-decoration: underline;
}

.content {
    display: flex;
    flex-direction: column;
    justify-content: start;
    align-items: center;
}

.btnLink {
    color: #202E44;
    text-align: center;
    font-size: 16px;
    font-style: normal;
    font-weight: normal;
    line-height: normal;
    display: flex;
    align-items: center;
    border-radius: 20px;
    border: 1px solid #ACB2BD;
    padding: 10px 16px;
    width: max-content;
    margin-top: 6px;
    transition: box-shadow .1s ease-out;
    width: 100 %;
    display: flex;
    justify-content: center;
    align-items: center;
    min-width: max-content;
}

.btnLink:hover {
    text-decoration: none;
    box-shadow: 00px 0px 5px gray;
}
```

```
* {
  box-sizing: border-box;
}

body {
  display: flex;
  flex-direction: column;
  justify-content: start;
  align-items: center;
}

.logo {
  width: 125px;
}

.logo:hover {
  cursor: pointer;
}

.content {
  width: 100 %;
  max-width: 1200px;
  display: flex;
  justify-content: start;
  align-items: center;
}

.inputText::placeholder {
    color: #99a1ad;
}

.inputText:hover::placeholder {
    color: #606c7f;
    border-color: #0a47f0;
}

.inputText {
  display: block;
  margin: auto;
  padding: 15px;
  border-style: solid;
  border-radius: 10px;
  border-width: 1px;
  border-color: #868f9e;
  width: 100 %;
  font-size: 20px;
  color: #202e44;
  max-width: 400px;
}

.inputText:hover {
    border-color: #0a47f0;
}

.inputText:focus {
    border-color: #0a47f0;
    outline: none;
    border-width: 2px;
}

input[type="text"][class=error],
input[type="password"][class=error] {
    border-color: #e30425;
}

/* Button Styles */
.submitButton {
  display: block;
  border: none;
  border-radius: 15px;
  background-color: #ff7800;
  color: white;
  margin: auto;
  max-width: 400px;
  width: 100 %;
  font-size: 1.2rem;
  padding: 15px 20px;
}

.submitButton:hover {
    background-color: #ff961f;
    box-shadow: 0 4px 4px 0px #c8c8c8;
    cursor: pointer;
}

.submitButton:active {
    background-color: #e45e00;
}

.form {
  box-sizing: border-box;
  display: flex;
  flex-direction: column;
  gap: 20px;
  padding: 10px 20px;
}
```

```
* {
    box-sizing: border-box;
}

/* Logo Styles */
.logo {
    width: 125px;
}

.logo:hover {
    cursor: pointer;
}

/* Input Styles */
.inputText::placeholder{
    color: #99a1ad;
}

.inputText:hover::placeholder {
    color: #606c7f;
    border-color: #0a47f0;
}


.inputText:hover {
    border-color: #0a47f0;
}

.inputText:focus {
    border-color: #0a47f0;
    outline: none;
    border-width: 2px;
}

input[type="text"][class=error],
input[type="password"][class=error] {
    border-color: #e30425;
}
```

```

.submitButton:hover {
    background-color: #ff961f;
    box-shadow: 0 4px 4px 0px #c8c8c8;
    cursor: pointer;
}

.submitButton:active {
    background-color: #e45e00;
}

.form {
    box-sizing: border-box;
    display: flex;
    flex-direction: column;
    gap: 20px;
    padding: 10px 20px;
    width: 100%;
    max-width: 1200px;
}

.inputText {
    display: block;
    margin: auto;
    padding: 15px;
    border-style: solid;
    border-radius: 10px;
    border-width: 1px;
    border-color: #868f9e;
    width: 100%;
    font-size: 20px;
    color: #202e44;
    max-width: 400px;
}

.submitButton {
    display: block;
    border: none;
    border-radius: 10px;
    background-color: #ff7800;
    color: white;
    margin: auto;
    max-width: 400px;
    width: 100%;
    font-size: 1.2rem;
    padding: 15px 20px;
}
```

```
/* Input Styles */
* {
    box-sizing: border-box;
  }
body {
    display: flex;
    flex-direction: column;
    justify-content: start;
    align-items: center;
}

.rigthTop {
    display: flex;
    justify-content: center;
    align-items: center;
    gap: 20px;
}

@media screen and (max-width: 700px) {
    .rigthTop {
        flex-direction: column;
    }
}

.content {
    width: 100 %;
    max-width: 1200px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    flex-direction: column;
    padding: 0px 20px;
}

.options {
    display: flex;
    margin: auto;
    justify-content: center;
}

/* Button Styles */
.submitButton {
    margin: auto;
    display: block;
    border: none;
    border-radius: 25px;
    background-color: #ff7800;
    color: white;
}

.submitButton:hover {
    background-color: #ff961f;
    box-shadow: 0 4px 4px 0px #c8c8c8;
    cursor: pointer;
}

.submitButton:active {
    background-color: #e45e00;
}

.logout {
    width: 150px;
    font-size: 15px;
    padding: 5px;
    display: block;
    border: none;
    border-radius: 10px;
    background-color: #ff7800;
    color: white;
}

.submit {
    width: 225px;
    font-size: 20px;
    padding: 5px;
}

/* Division Styles */
.movement {
    display: flex;
    justify-content: space-between;
    width: 100 %;
    max-width: 1400px;
}

.form {
    width: 100 %;
    max-width: 400px;
    display: flex;
    flex-direction: column;
    justify-content: start;
    align-items: center;
    gap: 20px;
}

.submitButton {
    display: block;
    border: none;
    border-radius: 10px;
    background-color: #ff7800;
    color: white;
    margin: auto;
    max-width: 400px;
    width: 100 %;
    font-size: 1.2rem;
    padding: 15px 20px;
}

.inputText {
    display: block;
    margin: auto;
    padding: 15px;
    border-style: solid;
    border-radius: 10px;
    border-width: 1px;
    border-color: #868f9e;
    width: 100 %;
    font-size: 20px;
    color: #202e44;
    max-width: 400px;
}
```

```
/* Input Styles */
input[type="text"]::placeholder,
input[type="password"]::placeholder,
input[type="date"]::placeholder {
    color: #99a1ad;
}

input[type="text"]:hover::placeholder,
input[type="password"]:hover::placeholder,
input[type="date"]:hover::placeholder {
    color: #606c7f;
    border-color: #0a47f0;
}

input[type="text"],
input[type="password"],
input[type="date"] {
    padding: 10px;
    width: 1225px;
    border-style: solid;
```

```css
    border-radius: 10px;
    border-width: 1px;
    border-color: #868f9e;
    font-size: 15px;
    color: #202e44;
}

input[type="text"]:hover,
input[type="password"]:hover,
input[type="date"]:hover {
    border-color: #0a47f0;
}

input[type="text"]:focus,
input[type="password"]:focus,
input[type="date"]:focus {
    border-color: #0a47f0;
    outline: none;
    border-width: 2px;
}

input[type="text"][class="error"],
input[type="password"][class="error"],
input[type="date"][class="error"] {
    border-color: #e30425;
}

.options {
    display: flex;
    justify-content: center;
}

/* Button Styles */
input[type="button"] {
    margin: auto;
    display: block;
    border: none;
    border-radius: 25px;
    background-color: #ff7800;
    color: white;
}

input[type="button"]:hover {
    background-color: #ff961f;
    box-shadow: 0 4px 4px 0px #c8c8c8;
    cursor: pointer;
}

input[type="button"]:active {
    background-color: #e45e00;
}

.logout {
    width: 150px;
    font-size: 15px;
    padding: 5px;
}

.submit {
    width: 225px;
    font-size: 20px;
    padding: 5px;
}

```

```
.menu {
    display: flex;
    flex-direction: column;
    gap: 20px;
    align-items: center;
    justify-content: center;
    min-height: 50vh;
}

.btnLink {
    color: #202E44;
    text-align: center;
    font-size: 18px;
    font-style: normal;
    font-weight: normal;
    line-height: normal;
    display: flex;
    align-items: center;
    border-radius: 20px;
    border: 1px solid #ACB2BD;
    padding: 10px 16px;
    width: max-content;
    margin-top: 6px;
    transition: box-shadow .1s ease-out;
    width: 100 %;
    display: flex;
    justify-content: center;
    align-items: center;
}

.btnLink:hover {
    text-decoration: none;
    box-shadow: 00px 0px 5px gray;
}
```

5. JavaScript:

```
addEventListener("load", getStatus);

function getStatus() {
    fetch2([], [], "estado_tarjeta.pl", function(response) {
        const balance = response.getElementsByTagName("balance")[0].childNodes[0].nodeValue;
            createBalanceElement(balance);

            const movements = response.getElementsByTagName("movement");
            for (let i = movements.length - 1; i >= 0; i--) {
                let amount = movements[i].getElementsByTagName("amount")[0].childNodes[0].nodeValue;
                let type = movements[i].getElementsByTagName("type")[0].childNodes[0].nodeValue;
                let date = movements[i].getElementsByTagName("date")[0].childNodes[0].nodeValue;
                createMovementElement(amount, type, date);
            }
    });
}

function createBalanceElement(balance) {
    const balanceElement = document.createElement("h4");
    balanceElement.innerHTML = balance;
    document.getElementById("balance").appendChild(balanceElement);
}

function createMovementElement(amount, type, date) {
    const movementElement = document.createElement("div");
    movementElement.classList.add("movement");

    const amountElement = document.createElement("h5");
    const typeElement = document.createElement("h5");
    const dateElement = document.createElement("h5");
    amountElement.innerHTML = amount;
    typeElement.innerHTML = type;
    dateElement.innerHTML = date;

    movementElement.appendChild(amountElement);
    movementElement.appendChild(typeElement);
    movementElement.appendChild(dateElement);

    document.getElementById("movements").appendChild(movementElement);
}
```

```
addEventListener("load", getStatus);

function getStatus() {
    fetch2([], [], "lista_tarjetas.pl", function (response) {
        const cards = response.getElementsByTagName("cards");
        for (let i = 0; i < cards.length; i++) {
            let id = cards[i].getElementsByTagName("id")[0].childNodes[0].nodeValue;
            let card_number = cards[i].getElementsByTagName("card_number")[0].childNodes[0].nodeValue;
            let code = cards[i].getElementsByTagName("code")[0].childNodes[0].nodeValue;
            let expire_date = cards[i].getElementsByTagName("expire_date")[0].childNodes[0].nodeValue;
            let currency = cards[i].getElementsByTagName("currency")[0].childNodes[0].nodeValue;
            createMovementElement(id, card_number, code, expire_date, currency);
        }
    });
}

function createMovementElement(amount, type, date) {
    const movementElement = document.createElement("div");
    movementElement.classList.add("movement");

    const amountElement = document.createElement("h5");
    const typeElement = document.createElement("h5");
    const dateElement = document.createElement("h5");
    amountElement.innerHTML = amount;
    typeElement.innerHTML = type;
    dateElement.innerHTML = date;

    movementElement.appendChild(amountElement);
    movementElement.appendChild(typeElement);
    movementElement.appendChild(dateElement);

    document.getElementById("movements").appendChild(movementElement);
}
```

```
function login() {
    fetch2(["user", "password"], ["type=cliente"], "login.pl", function(response) {
        const errors = response.getElementsByTagName("error");
        if (errors.length != 0) {
            createStatusElements(errors, "form");
        } else {
            window.location = "./index_clientes.html";
        }
    })
}
```

```
function login() {
    fetch2(["user", "password"], ["type=operario"], "login.pl", function(response) {
        const errors = response.getElementsByTagName("error");
        if (errors.length != 0) {
            createStatusElements(errors, "form");
        } else {
            window.location = "./index_operarios.html";
        }
    })
}
```

```
onload = function () {
    firstLog();
    updateTimer();
    setInterval(updateTimer, 1 * 1000);
};

function firstLog() {
    fetch2([], ["action=check", "type=cliente"], "sesion.pl", function(response) {
        let logged_in = response.getElementsByTagName("logged_in")[0].childNodes[0].nodeValue;
        if (logged_in == "1") {
            let welcome = document.createElement("h4");
            let name = response.getElementsByTagName("name")[0].childNodes[0].nodeValue;
            welcome.innerHTML = "Bienvenido, " + name + " ";
            document.getElementById("welcome").append(welcome);
        } else {
            redirectBack();
        }
    });
}

function updateTimer() {
    fetch2([], ["action=check", "type=cliente"], "sesion.pl", function(response) {
        let logged_in = response.getElementsByTagName("logged_in")[0].childNodes[0].nodeValue;
        if (logged_in == "1") {
            let timer = response.getElementsByTagName("expire_time")[0].childNodes[0].nodeValue;
            document.getElementById("timer").innerHTML = "Su sesión concluirá en: " + timer + "s";
        } else {
            redirectBack();
        }
    });
}

function closeSession() {
    fetch2([], ["action=close", "type=cliente"], "sesion.pl", function(response) {
        redirectBack();
    })
}

function redirectBack() {
    alert("Su sesión ha expirado");
    window.location = "./login_clientes.html";
}
```

```
onload = function () {
    firstLog();
    updateTimer();
    setInterval(updateTimer, 1 * 1000);
};

function firstLog() {
    fetch2([], ["action=check", "type=operario"], "sesion.pl", function(response) {
        let logged_in = response.getElementsByTagName("logged_in")[0].childNodes[0].nodeValue;
        if (logged_in == "1") {
            let welcome = document.createElement("h4");
            let name = response.getElementsByTagName("name")[0].childNodes[0].nodeValue;
            welcome.innerHTML = "Bienvenido, " + name + " ";
            document.getElementById("welcome").append(welcome);
        } else {
            redirectBack();
        }
    });
}

function updateTimer() {
    fetch2([], ["action=check", "type=operario"], "sesion.pl", function(response) {
        let logged_in = response.getElementsByTagName("logged_in")[0].childNodes[0].nodeValue;
        if (logged_in == "1") {
            let timer = response.getElementsByTagName("expire_time")[0].childNodes[0].nodeValue;
            document.getElementById("timer").innerHTML = "Su sesión concluirá en: " + timer + "s";
        } else {
            redirectBack();
        }
    });
}

function closeSession() {
    fetch2([], ["action=close", "type=operario"], "sesion.pl", function(response) {
        redirectBack();
    })
}

function redirectBack() {
    alert("Su sesión ha expirado");
    window.location = "./login_operarios.html";
}
```

```
function doMovement() {
    fetch2([["type", "deposit", "withdrawal", "transference"], "amount"], [], "movimiento_cliente.pl", function (response) {
        const errors = response.getElementsByTagName("error");
        if (errors.length != 0) {
            createStatusElements(errors, "form");
        } else {
            alert("Movimiento realizado correctamente");
            window.location = "./index_clientes.html";
        }
    });
}
```

```
function createClient() {
    fetch2(["dni", "name", "plastname", "mlastname", "bdate"], [], "registro_clientes.pl", function(response) {
        const errors = response.getElementsByTagName("error");
        if (errors.length != 0) {
            // TODO: update these when the v htmls are done
            createStatusElements(errors, "form");
        } else {
            alert("Cliente creado correctamente");
            window.location = "./index_operarios.html";
        }
    });
}
```

```
function createAccount() {
    // TODO: accounts are only created with these now
    fetch2(["user", "password", "dni"], [], "registro_cuentas.pl", function(response) {
        if (errors.length != 0) {
            // TODO: update these when the v htmls are done
            createStatusElements(errors, "form");
        } else {
            alert("Cuenta creada correctamente");
            window.location = "./index_operarios.html";
        }
    });
}
```

```
function createCard() {
    // TODO: cards are only created with these now
    fetch2(["number", "password", ["currency", "soles", "dolares"], "dni"], [], "registro_tarjetas.pl", function(response) {
        if (errors.length != 0) {
            // TODO: update these when the v htmls are done
            createStatusElements(errors, "form");
        } else {
            alert("Cuenta creada correctamente");
            location = "./index_operarios.html";
        }
    });
}

function generateNumber() {
    let number = "";
    for (let i = 0; i < 16; i++) {
        number += Math.floor(Math.random() * 9);
    }
    document.getElementById("number").value = number;
}

```

```
function generatePassword() {
    const chars = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
    let password = "";
    for (let i = 0; i < 10; i++) {
        password += chars.charAt(Math.random() * chars.length)
    }
    document.getElementById("password").value = password;
}
```

```
function redirect(url) {
    location = url;
}

// html_params son los parametros que se buscan en el formulario (el id tiene que tener el mismo nombre)
// se permite el ingreso de seleccion multiple con un arreglo, el primer elemento es el nombre del parametro y los siguientes las opciones a buscar en el formulario
// extra_params son parametros adicionales que puedes ingresar manualmente "type=operario"
// script es la ubicacion del archio perl, tiene que terminar en .pl
// func siempre debe tener el parametro (response)
function fetch2(html_params, extra_params, script, func) {
    let xhttp = new XMLHttpRequest();
    let query = "";
    for (let i = 0; i < html_params.length; i++) {
        try {
            if (typeof html_params[i] == "object") {
                let chosenInput;
                for (let j = 1; j < html_params[i].length; j++) {
                    let input = document.getElementById(html_params[i][j]);
                    if (input.checked) {
                        chosenInput = input;
                        break;
                    }
                }
                query += html_params[i][0] + "=" + chosenInput.value + "&";
            } else {
                let param_value = document.getElementById(html_params[i]).value;
                query += html_params[i] + "=" + param_value + "&";
            }
        } catch (err) {
            alert("Error obteniendo " + html_params[i]);
        }
    }

    for (let i = 0; i < extra_params.length; i++) {
        query += extra_params[i] + "&";
    }

    xhttp.onreadystatechange = function () {
        if (xhttp.readyState == 4 && xhttp.status == 200) {
            let response = xhttp.responseXML || new Document();
            func(response);
        }
    };

    xhttp.open("POST", "./cgi-bin/" + script, true);
    xhttp.setRequestHeader(’Content-Type’, ’application/x-www-form-urlencoded; charset=ISO-8859-1’);
    xhttp.send(query);
}

function removeStatusElements() {
    let errorElements = document.getElementsByName("error");
    while (errorElements.length > 0) {
        errorElements[0].remove();
    }
}

function createStatusElements(errors, parent) {
    removeStatusElements(parent);
    for (let i = 0; i < errors.length; i++) {
        let element = errors[i].getElementsByTagName("element")[0].firstChild.nodeValue;
        let message = errors[i].getElementsByTagName("message")[0].firstChild.nodeValue;
        createElement(element, message, parent);
    }
}

function createElement(element, message, parent) {
    const statusElement = document.createElement("h3");
    const placeElement = document.getElementById(element);

    placeElement.classList.add("error");
    statusElement.classList.add("error");

    statusElement.innerHTML = message;
    statusElement.setAttribute("name", "error");
    document.getElementById(parent).insertBefore(statusElement, placeElement);
}
```