

Código

1. Perl:

```
1  #!perl/bin/perl.exe
2
3  # Recibe: card_id
4  # Retorna:
5  # <status>
6  #     <movement>
7  #         <amount>cantidad</amount>
8  #         <type>tipo</type>
9  #         <date>fecha</date>
10 #     </movement>
11 #     <movement>
12 #         ...
13 #     </movement>
14 #     <balance>saldo</balance>
15 # </status>
16
17 use strict;
18 use warnings;
19 use CGI;
20 use CGI::Session;
21 use CGI::Cookie;
22 use DBI;
23
24 my %type_nums = (d => 1, w => 2, i => 3);
25 my %type_names = (1 => "DEPOSITO", 2 => "RETIRO", 3 => "TRANSFERENCIA ENVIADA", 4 => "TRANSFERENCIA RECIBIDA");
26
27 my $cgi = CGI->new;
28 $cgi->charset("UTF-8");
29
30 my %cookies = CGI::Cookie->fetch();
31 my $session_cookie = $cookies{"session_id_cliente"};
32
33 if ($session_cookie) {
34     my $session_id = $session_cookie->value();
35     my $session = CGI::Session->load($session_id);
36     my $card_id = 1;
37
38     my $u = "query";
39     my $p = "YR4AFJUC3nyRmasY";
40     my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
41     my $dbh = DBI->connect($dsn, $u, $p);
42
43     my $sth = $dbh->prepare("SELECT 'tarjetas'.moneda, 'movimientos'.monto, 'movimientos'.tipo,
44                             'movimientos'.fecha
45                             FROM tarjetas, movimientos
46                             WHERE tarjetas.id = '$card_id'");
47
48     $sth->execute;
49
50     my @row = $sth->fetchrow_array;
51     my $currency = $row[0] eq "s" ? "S/." : "\$";
52     my $amount = $currency.$row[1];
53     my $type = $type_names{$row[2]};
54     my $date = $row[3];
55     my $total = $row[1] * get_mult($row[2]);
56
57     print $cgi->header("text/xml");
58     print "<status>\n";
59     print_row($amount, $type, $date);
60 }
```

```

58     while (@row = $sth->fetchrow_array) {
59         $total = $total + ($row[1] * get_mult($row[2]));
60         my $amount = $currency.$row[1];
61         my $type = $type_names{$row[2]};
62         my $date = $row[3];
63         print_row($amount, $type, $date);
64     }
65     my $balance = $currency.$total;
66     print<<XML;
67         <balance>$balance</balance>
68     </status>
69 XML
70 }
71
72 sub get_mult {
73     my $mult = 1;
74     if ($type_nums{$_[0]} == 2 || $type_nums{$_[0]} == 3) {
75         $mult = -1;
76     }
77     return $mult;
78 }
79
80 sub print_row {
81     my $amount = $_[0];
82     my $type = $_[1];
83     my $date = $_[2];
84     print<<XML;
85         <movement>
86             <amount>$amount</amount>
87             <type>$type</type>
88             <date>$date</date>
89         </movement>
90 XML
91 }

```

```

1  #!perl/bin/perl.exe
2
3  # Recibe: nada
4  # Retorna:
5  # <cards>
6  #     <card>
7  #         <id>id</id>
8  #         <card_number>numero de tarjeta</card_number>
9  #         <code>codigo</code>
10 #         <expire_date>fecha de expiracion</expire_date>
11 #         <currency>moneda</currency>
12 #     </card>
13 #     <card>
14 #         ...
15 #     </card>
16 # </cards>
17
18 use strict;
19 use warnings;
20 use CGI;
21 use CGI::Session;
22 use CGI::Cookie;
23 use DBI;
24
25 my $cgi = CGI->new;
26 $cgi->charset("UTF-8");
27
28 my %cookies = CGI::Cookie->fetch();

```

```

29 my $session_cookie = $cookies{"session_id_cliente"};
30
31 if ($session_cookie) {
32     my $session_id = $session_cookie->value();
33     my $session = CGI::Session->load($session_id);
34     my $account_id = $session->param("account_id");
35
36     my $u = "query";
37     my $p = "YR4AFJUC3nyRmasY";
38     my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
39     my $dbh = DBI->connect($dsn, $u, $p);
40
41     my $sth = $dbh->prepare("SELECT 'id', 'numero', 'clave', 'vencimiento', 'moneda'
42                             FROM tarjetas
43                             WHERE cuenta_id = '$account_id'");
44     $sth->execute;
45
46     while (my @row = $sth->fetchrow_array) {
47         my $id = $row[0];
48         my $card_number = $row[1];
49         my $code = $row[2];
50         my $expire_date = $row[3];
51         my $currency = $row[4] eq "s" ? "SOLES" : "DÓLARES";
52
53         print $cgi->header("text/xml");
54         print "<errors>\n";
55         print_card($id, $card_number, $code, $expire_date, $currency);
56         print "</errors>\n";
57     }
58 }
59
60 sub print_card {
61     print<<XML;
62     <card>
63         <id>$_[0]</id>
64         <card_number>$_[1]</card_number>
65         <code>$_[2]</code>
66         <expire_date>$_[3]</expire_date>
67         <currency>$_[4]</currency>
68     </card>
69 XML
70 }

```

```

1  #!perl/bin/perl.exe
2
3  # Recibe: user, password, type (cliente u operario)
4  # Retorna:
5  # <errors>
6  #     <error>
7  #         <element>elemento</element>
8  #         <message>mensaje</message>
9  #     </error>
10 #     <error>
11 #         ...
12 #     </error>
13 # </errors>
14 # o solo la creacion de una cookie si todo esta bien :>
15
16 use strict;
17 use warnings;
18 use CGI;
19 use CGI::Session;
20 use CGI::Cookie;

```

```

21 use DBI;
22
23 my $cgi = CGI->new;
24 $cgi->charset("UTF-8");
25 my $user = $cgi->param("user");
26 my $password = $cgi->param("password");
27 my $type = $cgi->param("type");
28 my $session_time = 600;
29
30 my $user_status = check_user($user);
31 my $password_status = check_password($password);
32 my $type_status = check_type($type);
33 my %errors = (user => $user_status, password => $password_status, type => $type_status);
34
35 login();
36
37 sub login {
38     if (!$user_status && !$password_status && !$type_status) {
39         my $u = "query";
40         my $p = "YR4AFJUC3nyRmasY";
41         my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
42         my $dbh = DBI->connect($dsn, $u, $p);
43
44         my $query;
45         if ($type eq "cliente") {
46             $query = "SELECT 'cuentas'.'.id', 'clientes'.'.id', CONCAT('clientes'.'.nombres', ' ', 'clientes'.'.paterno',
47                 ' ', 'clientes'.'.materno')
48                 FROM cuentas, clientes
49                 WHERE cuentas.usuario = '$user' AND cuentas.clave = '$password' AND cuentas.cliente_id =
50                     clientes.id";
51         } else {
52             $query = "SELECT 'nombre' FROM operarios WHERE usuario='$user' AND clave='$password'";
53         }
54         my $sth = $dbh->prepare($query);
55         $sth->execute;
56
57         my @person = $sth->fetchrow_array;
58         if (@person) {
59             my $session = CGI::Session->new();
60             if ($type eq "cliente") {
61                 $session->param("account_id", $person[0]);
62                 $session->param("client_id", $person[1]);
63                 $session->param("name", $person[2]);
64             } else {
65                 $session->param("name", $person[0]);
66             }
67             $session->expire(time + $session_time);
68             $session->flush();
69
70             my $cookie = $cgi->cookie(-name => "session_id_$type",
71                 -value => $session->id(),
72                 -expires => time + $session_time,
73                 "-max-age" => time + $session_time);
74
75             $cookie->bake;
76
77             return;
78         } else {
79             $errors{login} = "El usuario y la clave no coinciden.";
80         }
81     }
82
83     print_errors();
84 }

```

```

83 sub check_user {
84     my $card_number = $_[0];
85     my $length = length($card_number);
86     if (!$card_number || $length == 0) {
87         return "Ingrese un usuario.";
88     }
89     if ($length > 30) {
90         return "Usuario muy largo.";
91     }
92 }
93
94 sub check_password {
95     my $password = $_[0];
96     my $length = length($password);
97     if (!$password || $length == 0) {
98         return "Ingrese una clave.";
99     }
100    if ($length > 30) {
101        return "Clave muy larga.";
102    }
103 }
104
105 sub check_type {
106     my $type = $_[0];
107     if ($type ne "cliente" && $type ne "operario") {
108         return "Tipo no valido.";
109     }
110 }
111
112 sub print_errors {
113     print $cgi->header("text/xml");
114     print "<errors>\n";
115     for my $key (keys %errors) {
116         if ($errors{$key}) {
117             print<<XML;
118             <error>
119                 <element>$key</element>
120                 <message>$errors{$key}</message>
121             </error>
122 XML
123         }
124     }
125     print "</errors>\n";
126 }

```

```

1  #!perl/bin/perl.exe
2
3  # Recibe: card_id, type (d, w, i), amount, other_card (only for i)
4  # deposit, withdrawal, transferin
5  # Retorna:
6  # <errors>
7  #     <error>
8  #         <element>elemento</element>
9  #         <message>mensaje</message>
10 #     </error>
11 #     <error>
12 #         ...
13 #     </error>
14 # </errors>
15 # o nada si todo esta bien :>
16
17 use strict;
18 use warnings;

```

```

19 use CGI;
20 use CGI::Session;
21 use CGI::Cookie;
22 use DBI;
23
24 my %type_nums = (d => 1, w => 2, i => 3);
25
26 my $cgi = CGI->new;
27 $cgi->charset("UTF-8");
28 my $card_id = "032";
29 my $type = $cgi->param("type");
30 my $amount = $cgi->param("amount");
31
32 my %cookies = CGI::Cookie->fetch();
33 my $session_cookie = $cookies{"session_id_cliente"};
34
35 my $u = "query";
36 my $p = "YR4AFJUC3nyRmasY";
37 my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
38 my $dbh = DBI->connect($dsn, $u, $p);
39
40 my $card_id_status = check_card_id($card_id);
41 my $type_status = check_type($type);
42 my $amount_status = check_amount($amount);
43
44 my %errors = (card_id => $card_id_status, type => $type_status, amount => $amount_status);
45
46 if ($session_cookie) {
47     my $session_id = $session_cookie->value();
48     my $session = CGI::Session->load($session_id);
49     my $account_id = $session->param("account_id");
50
51     transaction();
52 }
53
54 sub transaction {
55     if (!$card_id_status && !$amount_status && !$type_status) {
56         my $type_num = $type_nums{$type};
57         if ($type_num == 3) {
58             my $other_card = $cgi->param("other_card");
59             my $sth = $dbh->prepare("SELECT 'id' FROM tarjetas WHERE numero = '$other_card'");
60             $sth->execute;
61             my @row = $sth->fetchrow_array;
62             my $other_id = $row[0];
63
64             $sth = $dbh->prepare("INSERT INTO movimientos (tarjeta_id, monto, tipo) VALUES ($other_id, $amount, 4)");
65             $sth->execute;
66         }
67         my $sth = $dbh->prepare("INSERT INTO movimientos (tarjeta_id, monto, tipo) VALUES ($card_id, $amount, $type_num)");
68         $sth->execute;
69         print $cgi->header("text/xml");
70
71         return;
72     }
73
74     print_errors();
75 }
76
77 sub check_card_id {
78     my $card_id = $_[0];
79     my $length = length($card_id);
80     if (!$card_id || $length == 0) {
81         return "Tarjeta no valida.";
82     }
83 }

```

```

82     }
83 }
84
85 sub check_amount {
86     my $amount = $_[0];
87     if (!$amount) {
88         return "Ingrese una cantidad.";
89     }
90     if ($amount !~ /\d+$/ || $amount < 0) {
91         return "Cantidad no valida.";
92     }
93     if ($type eq "w") {
94         my $sth = $dbh->prepare("SELECT SUM(monto*tipo) FROM movimientos WHERE tarjeta_id = $card_id");
95         $sth->execute;
96         my @sum = $sth->fetchrow_array;
97         my $balance = $sum[0] || 0;
98         if ($balance - $amount < 0) {
99             return "Balance negativo."
100         }
101     }
102 }
103
104 sub check_type {
105     my $type = $_[0];
106     if (!$type) {
107         return "Marque un tipo";
108     }
109     if ($type !~ /^[dwi]$/) {
110         return "Tipo no valido";
111     }
112 }
113
114 sub print_errors {
115     print $cgi->header("text/xml");
116     print "<errors>\n";
117     for my $key (keys %errors) {
118         if ($errors{$key}) {
119             print<<XML;
120
121             <error>
122                 <element>$key</element>
123                 <message>$errors{$key}</message>
124             </error>
125
126             XML
127         }
128     }
129     print "</errors>\n";
130 }

```

```

1  #!/perl/bin/perl.exe
2
3  # Recibe: dni, name, plastname, mlastname, bdate
4  # Retorna:
5  # <errors>
6  #     <error>
7  #         <element>elemento</element>
8  #         <message>mensaje</message>
9  #     </error>
10 #     <error>
11 #         ...
12 #     </error>
13 # </errors>
14 # o nada si todo esta bien :>
15

```

```

16 use strict;
17 use warnings;
18 use CGI;
19 use CGI::Session;
20 use CGI::Cookie;
21 use DBI;
22
23 my $cgi = CGI->new;
24 $cgi->charset("UTF-8");
25 my $dni = $cgi->param("dni");
26 my $name = $cgi->param("name");
27 my $plastname = $cgi->param("plastname");
28 my $mlastname = $cgi->param("mlastname");
29 my $bdate = $cgi->param("bdate");
30
31 my $dni_status = check_DNI($dni);
32 my $name_status = check_name($name);
33 my $plastname_status = check_plastname($plastname);
34 my $mlastname_status = check_mlastname($mlastname);
35 my $bdate_status = check_bdate($bdate);
36
37 my %errors = (dni => $dni_status, name => $name_status, lastname => $plastname_status, mlastname =>
    $mlastname_status, bdate => $bdate_status);
38
39 register();
40
41 sub register {
42     if (!$dni_status && !$name_status && !$plastname_status && !$mlastname_status && !$bdate_status) {
43         my $u = "query";
44         my $p = "YR4AFJUC3nyRmasY";
45         my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
46         my $dbh = DBI->connect($dsn, $u, $p);
47
48         my $sth = $dbh->prepare("INSERT INTO clientes (dni, nombres, paterno, materno, nacimiento) VALUES ($dni,
            $name, $plastname, $mlastname, $bdate)");
49         $sth->execute();
50         print $cgi->header("text/xml");
51
52         return;
53     }
54
55     print_errors();
56 }
57
58 sub check_DNI {
59     my $dni = $_[0];
60     if (!$dni) {
61         return "Ingrese un DNI.";
62     }
63     if ($dni !~ /\d{8}$/) {
64         return "DNI no valido.";
65     }
66 }
67
68 sub check_name {
69     my $name = $_[0];
70     if (!$name) {
71         return "Ingrese un nombre.";
72     }
73     if (length($name) > 40) {
74         return "Nombre muy largo.";
75     }
76     if ($name !~ /\w{1,40}$/) {
77         return "Nombre no valido.";

```



```

78     }
79 }
80
81 sub check_plastname {
82     my $plastname = $_[0];
83     my $length = length($plastname);
84     if (!$plastname) {
85         return "Ingrese un apellido paterno.";
86     }
87     if ($length > 30) {
88         return "Apellido paterno muy largo.";
89     }
90     if ($plastname !~ /\w{1,30}$/) {
91         return "Apellido paterno no valido.";
92     }
93 }
94
95 sub check_mlastname {
96     my $mlastname = $_[0];
97     my $length = length($mlastname);
98     if (!$mlastname) {
99         return "Ingrese un apellido materno.";
100    }
101    if ($length > 30) {
102        return "Apellido materno muy largo.";
103    }
104    if ($mlastname !~ /\w{1,30}$/) {
105        return "Apellido materno no valido.";
106    }
107 }
108
109 sub check_bdate {
110     my $bdate = $_[0];
111     if (!$bdate) {
112         return "Ingrese una fecha de nacimiento.";
113     }
114     if ($bdate !~ /\d{4}\-\d{2}\-\d{2}$/) {
115         return "Fecha de nacimiento no valida.";
116     }
117 }
118
119 sub print_errors {
120     print $cgi->header("text/xml");
121     print "<errors>\n";
122     for my $key (keys %errors) {
123         if ($errors{$key}) {
124             print<<XML;
125             <error>
126                 <element>$key</element>
127                 <message>$errors{$key}</message>
128             </error>
129 XML
130         }
131     }
132     print "</errors>\n";
133 }

```

```

1  #!perl/bin/perl.exe
2
3  # Recibe: user, password, dni
4  # Retorna:
5  # <errors>
6  #     <error>

```

```

7  #         <element>elemento</element>
8  #         <message>mensaje</message>
9  #     </error>
10 #     <error>
11 #         ...
12 #     </error>
13 # </errors>
14 # o nada si todo esta bien :>
15
16 use strict;
17 use warnings;
18 use CGI;
19 use CGI::Session;
20 use CGI::Cookie;
21 use DBI;
22
23 my $cgi = CGI->new;
24 $cgi->charset("UTF-8");
25 my $user = $cgi->param("user");
26 my $password = $cgi->param("password");
27 my $dni = $cgi->param("dni");
28
29 our $client_id;
30 my $user_status = check_user($user);
31 my $password_status = check_password($password);
32 my $dni_status = check_DNI($dni);
33
34 my %errors = (user => $user_status, password => $password_status, dni => $dni_status);
35
36 register();
37
38 sub register {
39     if (!$user_status && !$password_status && !$dni_status) {
40         my $u = "query";
41         my $p = "YR4AFJUC3nyRmasY";
42         my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
43         my $dbh = DBI->connect($dsn, $u, $p);
44
45         my $sth = $dbh->prepare("INSERT INTO cuentas (usuario, clave, cliente_id) VALUES ($user, $password,
46             $client_id)");
47         $sth->execute;
48         print $cgi->header("text/xml");
49
50         return;
51     }
52
53     print_errors();
54 }
55
56 sub check_user {
57     my $card_number = $_[0];
58     my $length = length($card_number);
59     if (!$card_number || $length == 0) {
60         return "Ingrese un usuario.";
61     }
62     if ($length > 30) {
63         return "Usuario muy largo.";
64     }
65 }
66
67 sub check_password {
68     my $password = $_[0];
69     my $length = length($password);
70     if (!$password || $length == 0) {

```

```

70     return "Ingreso una clave.";
71 }
72 if ($length > 30) {
73     return "Clave muy larga.";
74 }
75 }
76
77 sub check_DNI {
78     my $dni = $_[0];
79     if (!$dni) {
80         return "Ingreso un DNI.";
81     }
82     if ($dni !~ /\d{8}$/) {
83         return "DNI no valido.";
84     }
85     my $sth = $dbh->prepare("SELECT 'id' FROM clientes WHERE dni = $dni");
86     $sth->execute;
87     my @row = $sth->fetchrow_array;
88     if (!@row) {
89         return "Cliente no existente.";
90     }
91     $client_id = $row[0];
92 }
93
94 sub print_errors {
95     print $cgi->header("text/xml");
96     print "<errors>\n";
97     for my $key (keys %errors) {
98         if ($errors{$key}) {
99             print<<XML;
100             <error>
101                 <element>$key</element>
102                 <message>$errors{$key}</message>
103             </error>
104             XML
105             }
106         }
107     print "</errors>\n";
108 }

```

```

1  #!perl/bin/perl.exe
2
3  # Recibe: number, password, currency (s o d), dni
4  # Retorna:
5  # <errors>
6  #     <error>
7  #         <element>elemento</element>
8  #         <message>mensaje</message>
9  #     </error>
10 #     <error>
11 #         ...
12 #     </error>
13 # </errors>
14 # o nada si todo esta bien :>
15
16 use strict;
17 use warnings;
18 use CGI;
19 use CGI::Session;
20 use CGI::Cookie;
21 use DBI;
22 use DateTime;
23

```

```

24 my $cgi = CGI->new;
25 $cgi->charset("UTF-8");
26 my $number = $cgi->param("number");
27 my $password = $cgi->param("password");
28 my $currency = $cgi->param("currency");
29 my $dni = $cgi->param("dni");
30 my $current_date = DateTime->now;
31 my $expire_date = $current_date->add(years => 7)->ymd;
32
33 our $account_id;
34 my $number_status = check_number($number);
35 my $password_status = check_password($password);
36 my $currency_status = check_currency($currency);
37 my $dni_status = check_DNI($dni);
38
39 register();
40
41 sub register {
42     if (!$number_status && !$password_status && !$currency_status && !$dni_status) {
43         my $u = "query";
44         my $p = "YR4AFJUC3nyRmasY";
45         my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
46         my $dbh = DBI->connect($dsn, $u, $p);
47
48         my $sth = $dbh->prepare("INSERT INTO tarjetas (numero, clave, vencimiento, cuenta_id) VALUES (?, ?, ?)");
49         $sth->execute($number, $password, $expire_date);
50         print $cgi->header("text/xml");
51
52         return;
53     }
54
55     print_errors();
56 }
57
58 sub check_number {
59     my $number = $_[0];
60     if (!$number) {
61         return "Ingrese un número de tarjeta.";
62     }
63     if ($number !~ /\d{16}$/) {
64         return "Número de tarjeta no valido.";
65     }
66 }
67
68 sub check_password {
69     my $password = $_[0];
70     if (!$password) {
71         return "Ingrese una clave.";
72     }
73     if (length($password) > 30) {
74         return "Clave muy larga.";
75     }
76     if (length($password) < 8) {
77         return "Clave muy corta.";
78     }
79 }
80
81 sub check_currency {
82     my $currency = $_[0];
83     if (!$currency) {
84         return "Marque una moneda.";
85     }
86     if ($currency !~ /^[sd]$/) {
87         return "Moneda no valida.";

```

```

88     }
89 }
90
91 sub check_DNI {
92     my $dni = $_[0];
93     if (!$dni) {
94         return "Ingrese un DNI.";
95     }
96     if ($dni !~ /\d{8}$/) {
97         return "DNI no valido.";
98     }
99     my $sth = $dbh->prepare("SELECT 'cuenta'. 'id'
100                             FROM cuentas, clientes
101                             WHERE clientes.dni = '$dni' AND cuentas.cliente_id = 'clientes_id'");
102     $sth->execute;
103     my @row = $sth->fetchrow_array;
104     if (!@row) {
105         return "Cliente no existente."
106     }
107     $client_id = $row[0];
108 }
109
110 sub print_errors {
111     print $cgi->header("text/xml");
112     print "<errors>\n";
113     for my $key (keys %errors) {
114         if ($errors{$key}) {
115             print<<XML;
116             <error>
117                 <element>$key</element>
118                 <message>$errors{$key}</message>
119             </error>
120             XML
121             }
122         }
123     print "</errors>\n";
124 }

```

```

1  #!/perl/bin/perl.exe
2
3  # Recibe: action (check [revisa si la sesion esta activa], close [cierra la sesion]), type (cliente u operario)
4  # Retorna:
5  # <session>
6  #     <logged_in>0 o 1</logged_in>
7  #     <name>nombre</name>
8  #     <expire_time>tiempo restante</expire_time>
9  # </session>
10
11 use strict;
12 use warnings;
13 use CGI;
14 use CGI::Session;
15 use CGI::Cookie;
16
17 my $cgi = CGI->new;
18 $cgi->charset("UTF-8");
19 my $action = $cgi->param("action");
20 my $type = $cgi->param("type");
21
22 my %cookies = CGI::Cookie->fetch();
23 my $session_cookie = $cookies{"session_id_$type"};
24
25 print ($cgi->header("text/xml"));

```

```

26 print "<session>\n";
27 if ($action eq "check") {
28     if ($session_cookie) {
29         my $session_id = $session_cookie->value();
30         my $session = CGI::Session->load($session_id);
31         my $name = $session->param("name");
32         my $expire_time = $session->expires() - time;
33         print<<XML;
34         <logged_in>1</logged_in>
35         <name>$name</name>
36         <expire_time>$expire_time</expire_time>
37     XML
38     } else {
39         print "<logged_in>0</logged_in>\n";
40     }
41 } elsif ($action eq "close") {
42     print "<logged_in>0</logged_in>\n";
43     if ($session_cookie) {
44         my $session_id = $session_cookie->value();
45         my $session = CGI::Session->load($session_id);
46         $session_cookie->expires("-1h");
47         $session->delete();
48         $session->flush();
49     }
50 }
51 print "</session>";

```

2. SQL:

```
1  -- phpMyAdmin SQL Dump
2  -- version 5.2.1
3  -- https://www.phpmyadmin.net/
4  --
5  -- Servidor: 127.0.0.1
6  -- Tiempo de generación: 17-01-2024 a las 04:43:38
7  -- Versión del servidor: 10.4.28-MariaDB
8  -- Versión de PHP: 8.2.4
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 START TRANSACTION;
12 SET time_zone = "+00:00";
13
14
15 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18 /*!40101 SET NAMES utf8mb4 */;
19
20 --
21 -- Base de datos: 'bancafinal'
22 --
23
24 -----
25
26 --
27 -- Estructura de tabla para la tabla 'clientes'
28 --
29
30 CREATE TABLE 'clientes' (
31   'id' int(11) NOT NULL,
32   'dni' varchar(8) NOT NULL,
33   'nombres' varchar(40) NOT NULL,
34   'paterno' varchar(30) NOT NULL,
35   'materno' varchar(30) NOT NULL,
36   'nacimiento' date NOT NULL,
37   'creado' timestamp NOT NULL DEFAULT current_timestamp(),
38   'estado' tinyint(1) NOT NULL DEFAULT 1
39 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
40
41 --
42 -- Volcado de datos para la tabla 'clientes'
43 --
44
45 INSERT INTO 'clientes' ('id', 'dni', 'nombres', 'paterno', 'materno', 'nacimiento', 'creado', 'estado') VALUES
46 (1, '54587912', 'Christian Raul', 'Mestas', 'Zegarra', '2006-04-17', '2024-01-17 02:19:50', 1);
47
48 -----
49
50 --
51 -- Estructura de tabla para la tabla 'cuentas'
52 --
53
54 CREATE TABLE 'cuentas' (
55   'id' int(11) NOT NULL,
56   'usuario' varchar(30) NOT NULL,
57   'clave' varchar(30) NOT NULL,
58   'creado' timestamp NOT NULL DEFAULT current_timestamp(),
59   'estado' tinyint(1) NOT NULL DEFAULT 1,
60   'cliente_id' int(11) NOT NULL
61 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```

62
63 --
64 -- Volcado de datos para la tabla 'cuentas'
65 --
66
67 INSERT INTO 'cuentas' ('id', 'usuario', 'clave', 'creado', 'estado', 'cliente_id') VALUES
68 (1, 'cmestasz', '123456789', '2024-01-17 02:20:14', 1, 1);
69
70 -----
71
72 --
73 -- Estructura de tabla para la tabla 'movimientos'
74 --
75
76 CREATE TABLE 'movimientos' (
77     'id' int(11) NOT NULL,
78     'tarjeta_id' int(11) NOT NULL,
79     'monto' float NOT NULL,
80     'tipo' tinyint(4) NOT NULL,
81     'fecha' timestamp NOT NULL DEFAULT current_timestamp()
82 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
83
84 --
85 -- Volcado de datos para la tabla 'movimientos'
86 --
87
88 INSERT INTO 'movimientos' ('id', 'tarjeta_id', 'monto', 'tipo', 'fecha') VALUES
89 (3, 1, 50000, 1, '2024-01-17 02:21:50'),
90 (5, 1, 25000, -1, '2024-01-17 02:23:23');
91
92 -----
93
94 --
95 -- Estructura de tabla para la tabla 'operarios'
96 --
97
98 CREATE TABLE 'operarios' (
99     'id' int(11) NOT NULL,
100     'nombre' varchar(30) NOT NULL,
101     'usuario' varchar(30) NOT NULL,
102     'clave' varchar(30) NOT NULL,
103     'creado' timestamp NOT NULL DEFAULT current_timestamp(),
104     'estado' tinyint(1) NOT NULL DEFAULT 1
105 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
106
107 -----
108
109 --
110 -- Estructura de tabla para la tabla 'tarjetas'
111 --
112
113 CREATE TABLE 'tarjetas' (
114     'id' int(11) NOT NULL,
115     'numero' varchar(16) NOT NULL,
116     'clave' varchar(30) NOT NULL,
117     'creado' timestamp NOT NULL DEFAULT current_timestamp(),
118     'vencimiento' datetime NOT NULL,
119     'sellado' tinyint(1) NOT NULL DEFAULT 1,
120     'estado' tinyint(1) NOT NULL DEFAULT 1,
121     'moneda' varchar(1) NOT NULL,
122     'cuenta_id' int(11) NOT NULL
123 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
124
125 --

```



```

126 -- Volcado de datos para la tabla 'tarjetas'
127 --
128
129 INSERT INTO 'tarjetas' ('id', 'numero', 'clave', 'creado', 'vencimiento', 'sellado', 'estado', 'moneda',
130     'cuenta_id') VALUES
131
132 (1, '5565894512347787', '123456789', '2024-01-17 02:20:32', '2024-01-17 03:20:19', 1, 1, 's', 1);
133
134 --
135
136 --
137 -- Índices para tablas volcadas
138 --
139 ALTER TABLE 'clientes'
140     ADD PRIMARY KEY ('id'),
141     ADD UNIQUE KEY 'dni' ('dni');
142
143 --
144 -- Índices de la tabla 'cuentas'
145 --
146 ALTER TABLE 'cuentas'
147     ADD PRIMARY KEY ('id'),
148     ADD KEY 'cliente_id' ('cliente_id') USING BTREE;
149
150 --
151 -- Índices de la tabla 'movimientos'
152 --
153 ALTER TABLE 'movimientos'
154     ADD PRIMARY KEY ('id'),
155     ADD KEY 'tarjeta_id' ('tarjeta_id') USING BTREE;
156
157 --
158 -- Índices de la tabla 'operarios'
159 --
160 ALTER TABLE 'operarios'
161     ADD PRIMARY KEY ('id');
162
163 --
164 -- Índices de la tabla 'tarjetas'
165 --
166 ALTER TABLE 'tarjetas'
167     ADD PRIMARY KEY ('id'),
168     ADD UNIQUE KEY 'numero' ('numero'),
169     ADD KEY 'cuenta_id' ('cuenta_id') USING BTREE;
170
171 --
172 -- AUTO_INCREMENT de las tablas volcadas
173 --
174
175 --
176 -- AUTO_INCREMENT de la tabla 'clientes'
177 --
178 ALTER TABLE 'clientes'
179     MODIFY 'id' int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
180
181 --
182 -- AUTO_INCREMENT de la tabla 'cuentas'
183 --
184 ALTER TABLE 'cuentas'
185     MODIFY 'id' int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
186
187 --
188 -- AUTO_INCREMENT de la tabla 'movimientos'

```

```

189  --
190  ALTER TABLE 'movimientos'
191      MODIFY 'id' int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;
192
193  --
194  -- AUTO_INCREMENT de la tabla 'operarios'
195  --
196  ALTER TABLE 'operarios'
197      MODIFY 'id' int(11) NOT NULL AUTO_INCREMENT;
198
199  --
200  -- AUTO_INCREMENT de la tabla 'tarjetas'
201  --
202  ALTER TABLE 'tarjetas'
203      MODIFY 'id' int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
204
205  --
206  -- Restricciones para tablas volcadas
207  --
208
209  --
210  -- Filtros para la tabla 'cuentas'
211  --
212  ALTER TABLE 'cuentas'
213      ADD CONSTRAINT 'cliente_id' FOREIGN KEY ('cliente_id') REFERENCES 'clientes' ('id') ON DELETE CASCADE ON UPDATE
          CASCADE;
214
215  --
216  -- Filtros para la tabla 'movimientos'
217  --
218  ALTER TABLE 'movimientos'
219      ADD CONSTRAINT 'movimientos_ibfk_1' FOREIGN KEY ('tarjeta_id') REFERENCES 'tarjetas' ('id') ON DELETE CASCADE ON
          UPDATE CASCADE;
220
221  --
222  -- Filtros para la tabla 'tarjetas'
223  --
224  ALTER TABLE 'tarjetas'
225      ADD CONSTRAINT 'tarjetas_ibfk_1' FOREIGN KEY ('cuenta_id') REFERENCES 'cuentas' ('id') ON DELETE CASCADE ON UPDATE
          CASCADE;
226  COMMIT;
227
228  /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
229  /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
230  /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
231
232  -- usuarios
233  CREATE USER 'query'@'%' IDENTIFIED BY 'YR4AFJUC3nyRmasY';
234  GRANT SELECT, INSERT ON 'bancafinal'.* TO 'query'@'%';

```

3. HTML:

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta charset="UTF-8" />
6     <title>Estado de cuenta</title>
7     <link rel="stylesheet" href="./css/estilos_generales.css" />
8     <link rel="stylesheet" href="./css/menus_clientes.css" />
9     <script src="./scripts/utilidades.js"></script>
10    <script src="./scripts/menus_clientes.js"></script>
11    <script src="./scripts/estado_tarjeta.js"></script>
12 </head>
13
14 <body>
15     <div id="content" class="content">
16         <div class="top">
17             <h2>Estado de cuenta</h2>
18             <div class="righthTop">
19                 <a href="./index_clientes.html" class="btnLink">Volver</a>
20                 <div>
21                     <h4 id="timer">Su sesión concluirá en:</h4>
22                     <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
23                 </div>
24             </div>
25         </div>
26         <div class="form">
27             <h3>Balance de cuenta:</h3>
28             <div id="balance"></div>
29             <h3>Últimos movimientos:</h3>
30         </div>
31         <div id="movements" style="width: 100%;">
32             <div class="movement">
33                 <h5 style="font-size: 20px;">CANTIDAD</h5>
34                 <h5 style="font-size: 20px;">TIPO</h5>
35                 <h5 style="font-size: 20px;">FECHA</h5>
36             </div>
37         </div>
38     </div>
39 </body>
40
41 </html>
```

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta charset="UTF-8" />
6     <title>Usuario</title>
7     <link rel="stylesheet" href="./css/estilos_generales.css" />
8     <link rel="stylesheet" href="./css/menus_clientes.css" />
9     <script src="./scripts/menus_clientes.js"></script>
10    <script src="./scripts/utilidades.js"></script>
11 </head>
12
13 <body>
14     <div id="content" class="content">
15         <div class="top">
16             <h2>
17                 Acceso rápido
18             <div id="welcome"></div>
```

```

19     </h2>
20     <div class="righthTop">
21         <a href="./movimiento_cliente.html" class="btnLink">Realizar un movimiento</a>
22         <a href="./estado_tarjeta.html" class="btnLink">Ver estado</a>
23         <div>
24             <h4 id="timer">Su sesión concluirá en:</h4>
25             <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
26         </div>
27     </div>
28 </div>
29 </div>
30 </body>
31
32 </html>

```

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta charset="UTF-8" />
6     <title>Usuario</title>
7     <link rel="stylesheet" href="./css/estilos_generales.css" />
8     <link rel="stylesheet" href="./css/menus_operarios.css" />
9     <script src="./scripts/utilidades.js"></script>
10    <script src="./scripts/menus_operarios.js"></script>
11 </head>
12
13 <body>
14     <div id="content" class="content">
15         <div class="top">
16             <h2>
17                 Acceso rápido
18                 <div id="welcome"></div>
19             </h2>
20             <div>
21                 <h4 id="timer">Su sesión concluirá en:</h4>
22                 <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
23             </div>
24         </div>
25         <div class="menu">
26             <a href="./registro_clientes.html" class="btnLink">Ingresar nuevo cliente</a>
27             <a href="./registro_cuentas.html" class="btnLink">Crear nueva cuenta</a>
28             <a href="./registro_tarjetas.html" class="btnLink">Crear nueva tarjeta</a>
29         </div>
30     </div>
31 </body>
32
33 </html>

```

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta charset="UTF-8" />
6     <title>Ingreso de Clientes</title>
7     <link rel="stylesheet" href="./css/estilos_generales.css" />
8     <link rel="stylesheet" href="./css/login_clientes.css" />
9     <script src="./scripts/utilidades.js"></script>
10    <script src="./scripts/login_clientes.js"></script>
11 </head>
12
13 <body>

```

```

14 <div id="content" class="content">
15   
16 </div>
17 <div id="form" class="form">
18   <h1>Ingreso de Clientes</h1>
19   <input type="text" id="user" class="inputText" placeholder="Usuario" required />
20   <input type="password" id="password" placeholder="Contraseña" class="inputText" />
21   <input type="button" class="submitButton" onclick="login()" value="Ingresar"/>
22 </div>
23 </body>
24
25 </html>

```

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8" />
6   <title>Ingreso de Operarios</title>
7   <link rel="stylesheet" href="./css/estilos_generales.css" />
8   <link rel="stylesheet" href="./css/login_operarios.css" />
9   <script src="./scripts/utilidades.js"></script>
10  <script src="./scripts/login_operarios.js"></script>
11 </head>
12
13 <body>
14   <div id="content" class="content">
15     
16     <h1>Ingreso de Operarios</h1>
17     <div id="form" class="form">
18       <input type="text" id="user" placeholder="Usuario" class="inputText" />
19       <input type="password" id="password" placeholder="Contraseña" class="inputText" />
20       <input type="button" id="login" class="submitButton" onclick="login()" value="Ingresar" />
21     </div>
22   </div>
23 </body>
24
25 </html>

```

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8" />
6   <title>Realizar un movimiento</title>
7   <link rel="stylesheet" href="./css/estilos_generales.css" />
8   <link rel="stylesheet" href="./css/menus_clientes.css" />
9   <script src="./scripts/utilidades.js"></script>
10  <script src="./scripts/menus_clientes.js"></script>
11  <script src="./scripts/movimiento_cliente.js"></script>
12 </head>
13
14 <body>
15   <div id="content" class="content">
16     <div class="top">
17       <h2>
18         Asistente de movimientos
19         <div id="welcome"></div>
20       </h2>
21       <div class="righTop">
22         <a href="./index_clientes.html" class="btnLink">Volver</a>
23       </div>

```

```

24         <h4 id="timer">Su sesión concluirá en:</h4>
25         <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
26     </div>
27 </div>
28 </div>
29
30
31 <div id="form" class="form">
32     <h3>Tipo de movimiento:</h3>
33     <div id="type" class="options">
34         <input type="radio" id="deposit" name="type" value="d" checked />
35         <label for="d">Depósito</label>
36         <input type="radio" id="withdrawal" name="type" value="w" />
37         <label for="w">Retiro</label>
38         <input type="radio" id="transference" name="type" value="t" />
39         <label for="t">Transferencia</label>
40     </div>
41     <input type="text" id="amount" placeholder="Cantidad" class="inputText" />
42     <input type="button" class="submitButton" onclick="doMovement()" value="Realizar movimiento" />
43 </div>
44 </div>
45 </body>
46
47 </html>

```

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta charset="UTF-8" />
6     <title>Creación de clientes</title>
7     <link rel="stylesheet" href="./css/estilos_generales.css" />
8     <link rel="stylesheet" href="./css/menus_operarios.css" />
9     <script src="./scripts/utilidades.js"></script>
10    <script src="./scripts/menus_operarios.js"></script>
11    <script src="./scripts/registro_clientes.js"></script>
12 </head>
13
14 <body>
15     <div id="content" class="content">
16         <div class="top">
17             <div id="welcome"></div>
18             <div>
19                 <h4 id="timer">Su sesión concluirá en:</h4>
20                 <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
21             </div>
22         </div>
23         <div style="height: 15px"></div>
24         <a href="./index_operarios.html">Volver</a>
25         <div style="height: 10px"></div>
26         <h2>Asistente de creación de clientes</h2>
27         <div style="height: 25px"></div>
28         <input type="text" id="dni" placeholder="DNI" />
29         <div style="height: 25px"></div>
30         <input type="text" id="name" placeholder="Nombres completos" />
31         <div style="height: 25px"></div>
32         <input type="text" id="pllastname" placeholder="Apellido paterno" />
33         <div style="height: 25px"></div>
34         <input type="text" id="mlastname" placeholder="Apellido materno" />
35         <div style="height: 25px"></div>
36         <input type="date" id="bdate" placeholder="Fecha de nacimiento" />
37         <div style="height: 25px"></div>
38         <input type="button" class="submit" onclick="createClient()" value="Crear cliente" />

```

```

39     </div>
40 </body>
41
42 </html>

```

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta charset="UTF-8" />
6     <title>Creación de cuentas</title>
7     <link rel="stylesheet" href="./css/estilos_generales.css" />
8     <link rel="stylesheet" href="./css/menus_operarios.css" />
9     <script src="./scripts/utilidades.js"></script>
10    <script src="./scripts/menus_operarios.js"></script>
11    <script src="./scripts/registro_cuentas.js"></script>
12 </head>
13
14 <body>
15     <div id="content" class="content">
16         <div class="top">
17             <div id="welcome"></div>
18             <div>
19                 <h4 id="timer">Su sesión concluirá en:</h4>
20                 <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
21             </div>
22         </div>
23         <div style="height: 15px"></div>
24         <a href="./index_operarios.html">Volver</a>
25         <div style="height: 10px"></div>
26         <h2>Asistente de creación de cuentas</h2>
27         <div style="height: 25px"></div>
28         <input type="text" id="number" placeholder="Número de tarjeta" />
29         <div style="height: 25px"></div>
30         <input type="button" class="submit" onclick="redirect('./registro_tarjetas.html')"
31             value="Creación de tarjetas" />
32         <div style="height: 25px"></div>
33         <h3>Tipo de moneda:</h3>
34         <div id="currency" class="options">
35             <input type="radio" id="soles" name="currency" value="s" checked />
36             <label for="soles">Soles</label>
37             <div style="width: 50px"></div>
38             <input type="radio" id="dollars" name="currency" value="d" />
39             <label for="dollars">Dólares</label>
40         </div>
41         <div style="height: 45px"></div>
42         <input type="text" id="dni" placeholder="DNI del cliente" />
43         <div style="height: 25px"></div>
44         <input type="button" class="submit" onclick="createAccount()" value="Crear cuenta" />
45     </div>
46 </body>
47
48 </html>

```

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta charset="UTF-8" />
6     <title>Creación de tarjetas</title>
7     <link rel="stylesheet" href="./css/estilos_generales.css" />
8     <link rel="stylesheet" href="./css/menus_operarios.css" />

```

```

9      <script src="./scripts/utilidades.js"></script>
10     <script src="./scripts/menus_operarios.js"></script>
11     <script src="./scripts/registro_tarjetas.js"></script>
12 </head>
13
14 <body>
15     <div id="content" class="content">
16         <div class="top">
17             <div id="welcome"></div>
18             <div>
19                 <h4 id="timer">Su sesión concluirá en:</h4>
20                 <input type="button" class="logout" onclick="closeSession()" value="Cerrar sesión" />
21             </div>
22         </div>
23         <div style="height: 15px"></div>
24         <a href="./index_operarios.html">Volver</a>
25         <div style="height: 10px"></div>
26         <h2>Asistente de creación de cuentas</h2>
27         <div style="height: 25px"></div>
28         <input type="button" class="submit" onclick="generateNumber()" value="Generar número" />
29         <div style="height: 25px"></div>
30         <input type="text" id="number" placeholder="Número de tarjeta" />
31         <div style="height: 25px"></div>
32         <input type="button" class="submit" onclick="generatePassword()" value="Generar clave" />
33         <div style="height: 25px"></div>
34         <input type="text" id="password" placeholder="Clave" />
35         <div style="height: 25px"></div>
36         <input type="button" class="submit" onclick="createCard()" value="Crear tarjeta" />
37     </div>
38 </body>
39
40 </html>

```

4. CSS:

```
1  /* Body Style */
2  body {
3      font-family: sans-serif;
4      background: linear-gradient(to top right, #d9ebfd, #fef9de);
5      min-height: 100vh;
6      margin: 0;
7      padding: 0;
8  }
9
10 /* Content Style */
11
12 .top {
13     box-sizing: border-box;
14     display: flex;
15     width: 100%;
16     max-width: 1400px;
17     justify-content: space-between;
18     align-items: center;
19     padding: 0px 40px;
20 }
21
22 @media screen and (max-width: 700px) {
23     .top {
24         flex-direction: column;
25     }
26 }
27
28 /* Header Styles */
29
30 /* Titles */
31 h1 {
32     text-align: center;
33     font-size: 40px;
34     color: #002a8d;
35 }
36
37 /* Subtitles */
38 h2 {
39     text-align: left;
40     font-size: 35px;
41     color: #002a8d;
42 }
43
44 /* Messages */
45 h3 {
46     text-align: center;
47     font-size: 20px;
48     color: #002a8d;
49 }
50
51 .error {
52     color: #e30425;
53 }
54
55 /* Text */
56 h4 {
57     text-align: center;
58     font-size: 15px;
59     color: #202e44;
60 }
61
```

```

62  /* Text - Inline */
63  h5 {
64      flex: 1;
65      display: inline-block;
66      text-align: center;
67      font-size: 15px;
68      color: #202e44;
69  }
70
71  /* Links */
72  a {
73      text-align: left;
74      font-size: 20px;
75      text-decoration: none;
76      color: #202e44;
77  }
78
79  a:hover {
80      text-decoration: underline;
81  }
82
83  .content {
84      display: flex;
85      flex-direction: column;
86      justify-content: start;
87      align-items: center;
88  }
89
90  .btnLink {
91      color: #202E44;
92      text-align: center;
93      font-size: 16px;
94      font-style: normal;
95      font-weight: normal;
96      line-height: normal;
97      display: flex;
98      align-items: center;
99      border-radius: 20px;
100     border: 1px solid #ACB2BD;
101     padding: 10px 16px;
102     width: max-content;
103     margin-top: 6px;
104     transition: box-shadow .1s ease-out;
105     width: 100%;
106     display: flex;
107     justify-content: center;
108     align-items: center;
109     min-width: max-content;
110 }
111
112 .btnLink:hover {
113     text-decoration: none;
114     box-shadow: 00px 0px 5px gray;
115 }

```

```

1  * {
2      box-sizing: border-box;
3  }
4
5  body {
6      display: flex;
7      flex-direction: column;
8      justify-content: start;

```

```

9     align-items: center;
10 }
11
12 .logo {
13     width: 125px;
14 }
15
16 .logo:hover {
17     cursor: pointer;
18 }
19
20 .content {
21     width: 100%;
22     max-width: 1200px;
23     display: flex;
24     justify-content: start;
25     align-items: center;
26 }
27
28 .inputText::placeholder {
29     color: #99a1ad;
30 }
31
32 .inputText:hover::placeholder {
33     color: #606c7f;
34     border-color: #0a47f0;
35 }
36
37 .inputText {
38     display: block;
39     margin: auto;
40     padding: 15px;
41     border-style: solid;
42     border-radius: 10px;
43     border-width: 1px;
44     border-color: #868f9e;
45     width: 100%;
46     font-size: 20px;
47     color: #202e44;
48     max-width: 400px;
49 }
50
51 .inputText:hover {
52     border-color: #0a47f0;
53 }
54
55 .inputText:focus {
56     border-color: #0a47f0;
57     outline: none;
58     border-width: 2px;
59 }
60
61 input[type="text"][class=error],
62 input[type="password"][class=error] {
63     border-color: #e30425;
64 }
65
66 /* Button Styles */
67 .submitButton {
68     display: block;
69     border: none;
70     border-radius: 15px;
71     background-color: #ff7800;
72     color: white;

```

```

73     margin: auto;
74     max-width: 400px;
75     width: 100%;
76     font-size: 1.2rem;
77     padding: 15px 20px;
78 }
79
80 .submitButton:hover {
81     background-color: #ff961f;
82     box-shadow: 0 4px 4px 0px #c8c8c8;
83     cursor: pointer;
84 }
85
86 .submitButton:active {
87     background-color: #e45e00;
88 }
89
90 .form {
91     box-sizing: border-box;
92     display: flex;
93     flex-direction: column;
94     gap: 20px;
95     padding: 10px 20px;
96 }

```

```

1  * {
2      box-sizing: border-box;
3  }
4
5  /* Logo Styles */
6  .logo {
7      width: 125px;
8  }
9
10 .logo:hover {
11     cursor: pointer;
12 }
13
14 /* Input Styles */
15 .inputText::placeholder{
16     color: #99a1ad;
17 }
18
19 .inputText:hover::placeholder {
20     color: #606c7f;
21     border-color: #0a47f0;
22 }
23
24
25 .inputText:hover {
26     border-color: #0a47f0;
27 }
28
29 .inputText:focus {
30     border-color: #0a47f0;
31     outline: none;
32     border-width: 2px;
33 }
34
35 input[type="text"][class=error],
36 input[type="password"][class=error] {
37     border-color: #e30425;
38 }

```

```

39
40 .submitButton:hover {
41     background-color: #ff961f;
42     box-shadow: 0 4px 4px 0px #c8c8c8;
43     cursor: pointer;
44 }
45
46 .submitButton:active {
47     background-color: #e45e00;
48 }
49
50 .form {
51     box-sizing: border-box;
52     display: flex;
53     flex-direction: column;
54     gap: 20px;
55     padding: 10px 20px;
56     width: 100%;
57     max-width: 1200px;
58 }
59
60 .inputText {
61     display: block;
62     margin: auto;
63     padding: 15px;
64     border-style: solid;
65     border-radius: 10px;
66     border-width: 1px;
67     border-color: #868f9e;
68     width: 100%;
69     font-size: 20px;
70     color: #202e44;
71     max-width: 400px;
72 }
73
74 .submitButton {
75     display: block;
76     border: none;
77     border-radius: 10px;
78     background-color: #ff7800;
79     color: white;
80     margin: auto;
81     max-width: 400px;
82     width: 100%;
83     font-size: 1.2rem;
84     padding: 15px 20px;
85 }

```

```

1  /* Input Styles */
2  * {
3      box-sizing: border-box;
4  }
5  body {
6      display: flex;
7      flex-direction: column;
8      justify-content: start;
9      align-items: center;
10 }
11
12 .rightTop {
13     display: flex;
14     justify-content: center;
15     align-items: center;

```

```

16     gap: 20px;
17 }
18
19 @media screen and (max-width: 700px) {
20     .righthTop {
21         flex-direction: column;
22     }
23 }
24
25 .content {
26     width: 100%;
27     max-width: 1200px;
28     display: flex;
29     justify-content: space-between;
30     align-items: center;
31     flex-direction: column;
32     padding: 0px 20px;
33 }
34
35 .options {
36     display: flex;
37     margin: auto;
38     justify-content: center;
39 }
40
41 /* Button Styles */
42 .submitButton {
43     margin: auto;
44     display: block;
45     border: none;
46     border-radius: 25px;
47     background-color: #ff7800;
48     color: white;
49 }
50
51 .submitButton:hover {
52     background-color: #ff961f;
53     box-shadow: 0 4px 4px #c8c8c8;
54     cursor: pointer;
55 }
56
57 .submitButton:active {
58     background-color: #e45e00;
59 }
60
61 .logout {
62     width: 150px;
63     font-size: 15px;
64     padding: 5px;
65     display: block;
66     border: none;
67     border-radius: 10px;
68     background-color: #ff7800;
69     color: white;
70 }
71
72 .submit {
73     width: 225px;
74     font-size: 20px;
75     padding: 5px;
76 }
77
78 /* Division Styles */
79 .movement {

```

```

80     display: flex;
81     justify-content: space-between;
82     width: 100%;
83     max-width: 1400px;
84 }
85
86 .form {
87     width: 100%;
88     max-width: 400px;
89     display: flex;
90     flex-direction: column;
91     justify-content: start;
92     align-items: center;
93     gap: 20px;
94 }
95
96 .submitButton {
97     display: block;
98     border: none;
99     border-radius: 10px;
100    background-color: #ff7800;
101    color: white;
102    margin: auto;
103    max-width: 400px;
104    width: 100%;
105    font-size: 1.2rem;
106    padding: 15px 20px;
107 }
108
109 .inputText {
110     display: block;
111     margin: auto;
112     padding: 15px;
113     border-style: solid;
114     border-radius: 10px;
115     border-width: 1px;
116     border-color: #868f9e;
117     width: 100%;
118     font-size: 20px;
119     color: #202e44;
120     max-width: 400px;
121 }

```

```

1  /* Input Styles */
2  input[type="text"]::placeholder,
3  input[type="password"]::placeholder,
4  input[type="date"]::placeholder {
5      color: #99a1ad;
6  }
7
8  input[type="text"]:hover::placeholder,
9  input[type="password"]:hover::placeholder,
10 input[type="date"]:hover::placeholder {
11     color: #606c7f;
12     border-color: #0a47f0;
13 }
14
15 input[type="text"],
16 input[type="password"],
17 input[type="date"] {
18     padding: 10px;
19     width: 1225px;
20     border-style: solid;

```

```

21     border-radius: 10px;
22     border-width: 1px;
23     border-color: #868f9e;
24     font-size: 15px;
25     color: #202e44;
26 }
27
28 input[type="text"]:hover,
29 input[type="password"]:hover,
30 input[type="date"]:hover {
31     border-color: #0a47f0;
32 }
33
34 input[type="text"]:focus,
35 input[type="password"]:focus,
36 input[type="date"]:focus {
37     border-color: #0a47f0;
38     outline: none;
39     border-width: 2px;
40 }
41
42 input[type="text"][class="error"],
43 input[type="password"][class="error"],
44 input[type="date"][class="error"] {
45     border-color: #e30425;
46 }
47
48 .options {
49     display: flex;
50     justify-content: center;
51 }
52
53 /* Button Styles */
54 input[type="button"] {
55     margin: auto;
56     display: block;
57     border: none;
58     border-radius: 25px;
59     background-color: #ff7800;
60     color: white;
61 }
62
63 input[type="button"]:hover {
64     background-color: #ff961f;
65     box-shadow: 0 4px 4px 0px #c8c8c8;
66     cursor: pointer;
67 }
68
69 input[type="button"]:active {
70     background-color: #e45e00;
71 }
72
73 .logout {
74     width: 150px;
75     font-size: 15px;
76     padding: 5px;
77 }
78
79 .submit {
80     width: 225px;
81     font-size: 20px;
82     padding: 5px;
83 }
84

```



```
85 .menu {
86     display: flex;
87     flex-direction: column;
88     gap: 20px;
89     align-items: center;
90     justify-content: center;
91     min-height: 50vh;
92 }
93
94 .btnLink {
95     color: #202E44;
96     text-align: center;
97     font-size: 18px;
98     font-style: normal;
99     font-weight: normal;
100    line-height: normal;
101    display: flex;
102    align-items: center;
103    border-radius: 20px;
104    border: 1px solid #ACB2BD;
105    padding: 10px 16px;
106    width: max-content;
107    margin-top: 6px;
108    transition: box-shadow .1s ease-out;
109    width: 100%;
110    display: flex;
111    justify-content: center;
112    align-items: center;
113 }
114
115 .btnLink:hover {
116     text-decoration: none;
117     box-shadow: 00px 0px 5px gray;
118 }
```

5. JavaScript:

```
1 addEventListener("load", getStatus);
2
3 function getStatus() {
4     fetch2([], [], "estado_tarjeta.pl", function(response) {
5         const balance = response.getElementsByTagName("balance")[0].childNodes[0].nodeValue;
6         createBalanceElement(balance);
7
8         const movements = response.getElementsByTagName("movement");
9         for (let i = movements.length - 1; i >= 0; i--) {
10             let amount = movements[i].getElementsByTagName("amount")[0].childNodes[0].nodeValue;
11             let type = movements[i].getElementsByTagName("type")[0].childNodes[0].nodeValue;
12             let date = movements[i].getElementsByTagName("date")[0].childNodes[0].nodeValue;
13             createMovementElement(amount, type, date);
14         }
15     });
16 }
17
18 function createBalanceElement(balance) {
19     const balanceElement = document.createElement("h4");
20     balanceElement.innerHTML = balance;
21     document.getElementById("balance").appendChild(balanceElement);
22 }
23
24 function createMovementElement(amount, type, date) {
25     const movementElement = document.createElement("div");
26     movementElement.classList.add("movement");
27
28     const amountElement = document.createElement("h5");
29     const typeElement = document.createElement("h5");
30     const dateElement = document.createElement("h5");
31     amountElement.innerHTML = amount;
32     typeElement.innerHTML = type;
33     dateElement.innerHTML = date;
34
35     movementElement.appendChild(amountElement);
36     movementElement.appendChild(typeElement);
37     movementElement.appendChild(dateElement);
38
39     document.getElementById("movements").appendChild(movementElement);
40 }

```

```
1 addEventListener("load", getStatus);
2
3 function getStatus() {
4     fetch2([], [], "lista_tarjetas.pl", function(response) {
5         const cards = response.getElementsByTagName("cards");
6         for (let i = 0; i < cards.length; i++) {
7             let id = cards[i].getElementsByTagName("id")[0].childNodes[0].nodeValue;
8             let card_number = cards[i].getElementsByTagName("card_number")[0].childNodes[0].nodeValue;
9             let code = cards[i].getElementsByTagName("code")[0].childNodes[0].nodeValue;
10            let expire_date = cards[i].getElementsByTagName("expire_date")[0].childNodes[0].nodeValue;
11            let currency = cards[i].getElementsByTagName("currency")[0].childNodes[0].nodeValue;
12            createMovementElement(id, card_number, code, expire_date, currency);
13        }
14    });
15 }
16
17 function createMovementElement(amount, type, date) {
18     const movementElement = document.createElement("div");
19     movementElement.classList.add("movement");

```

```

20
21     const amountElement = document.createElement("h5");
22     const typeElement = document.createElement("h5");
23     const dateElement = document.createElement("h5");
24     amountElement.innerHTML = amount;
25     typeElement.innerHTML = type;
26     dateElement.innerHTML = date;
27
28     movementElement.appendChild(amountElement);
29     movementElement.appendChild(typeElement);
30     movementElement.appendChild(dateElement);
31
32     document.getElementById("movements").appendChild(movementElement);
33 }

```

```

1  function login() {
2      fetch2(["user", "password"], ["type=cliente"], "login.pl", function(response) {
3          const errors = response.getElementsByTagName("error");
4          if (errors.length !== 0) {
5              createStatusElements(errors, "form");
6          } else {
7              window.location = "./index_clientes.html";
8          }
9      })
10 }

```

```

1  function login() {
2      fetch2(["user", "password"], ["type=operario"], "login.pl", function(response) {
3          const errors = response.getElementsByTagName("error");
4          if (errors.length !== 0) {
5              createStatusElements(errors, "form");
6          } else {
7              window.location = "./index_operarios.html";
8          }
9      })
10 }

```

```

1  onload = function () {
2      firstLog();
3      updateTimer();
4      setInterval(updateTimer, 1 * 1000);
5  };
6
7  function firstLog() {
8      fetch2([], ["action=check", "type=cliente"], "sesion.pl", function(response) {
9          let logged_in = response.getElementsByTagName("logged_in")[0].childNodes[0].nodeValue;
10         if (logged_in == "1") {
11             let welcome = document.createElement("h4");
12             let name = response.getElementsByTagName("name")[0].childNodes[0].nodeValue;
13             welcome.innerHTML = "Bienvenido, " + name + " ";
14             document.getElementById("welcome").append(welcome);
15         } else {
16             redirectBack();
17         }
18     });
19 }
20
21 function updateTimer() {
22     fetch2([], ["action=check", "type=cliente"], "sesion.pl", function(response) {
23         let logged_in = response.getElementsByTagName("logged_in")[0].childNodes[0].nodeValue;
24         if (logged_in == "1") {

```

```

25         let timer = response.getElementsByTagName("expire_time")[0].childNodes[0].nodeValue;
26         document.getElementById("timer").innerHTML = "Su sesión concluirá en: " + timer + "s";
27     } else {
28         redirectBack();
29     }
30 });
31 }
32
33 function closeSession() {
34     fetch2([], ["action=close", "type=cliente"], "sesion.pl", function(response) {
35         redirectBack();
36     })
37 }
38
39 function redirectBack() {
40     alert("Su sesión ha expirado");
41     window.location = "./login_clientes.html";
42 }

```

```

1  onload = function () {
2      firstLog();
3      updateTimer();
4      setInterval(updateTimer, 1 * 1000);
5  };
6
7  function firstLog() {
8      fetch2([], ["action=check", "type=operario"], "sesion.pl", function(response) {
9          let logged_in = response.getElementsByTagName("logged_in")[0].childNodes[0].nodeValue;
10         if (logged_in == "1") {
11             let welcome = document.createElement("h4");
12             let name = response.getElementsByTagName("name")[0].childNodes[0].nodeValue;
13             welcome.innerHTML = "Bienvenido, " + name + " ";
14             document.getElementById("welcome").append(welcome);
15         } else {
16             redirectBack();
17         }
18     });
19 }
20
21 function updateTimer() {
22     fetch2([], ["action=check", "type=operario"], "sesion.pl", function(response) {
23         let logged_in = response.getElementsByTagName("logged_in")[0].childNodes[0].nodeValue;
24         if (logged_in == "1") {
25             let timer = response.getElementsByTagName("expire_time")[0].childNodes[0].nodeValue;
26             document.getElementById("timer").innerHTML = "Su sesión concluirá en: " + timer + "s";
27         } else {
28             redirectBack();
29         }
30     });
31 }
32
33 function closeSession() {
34     fetch2([], ["action=close", "type=operario"], "sesion.pl", function(response) {
35         redirectBack();
36     })
37 }
38
39 function redirectBack() {
40     alert("Su sesión ha expirado");
41     window.location = "./login_operarios.html";
42 }

```

```

1 function doMovement() {
2     fetch2(["type", "deposit", "withdrawal", "transference", "amount"], [], "movimiento_cliente.pl", function
3         (response) {
4             const errors = response.getElementsByTagName("error");
5             if (errors.length != 0) {
6                 createStatusElements(errors, "form");
7             } else {
8                 alert("Movimiento realizado correctamente");
9                 window.location = "./index_clientes.html";
10             }
11         });
12 }

```

```

1 function createClient() {
2     fetch2(["dni", "name", "pllastname", "mlastname", "bdate"], [], "registro_clientes.pl", function(response) {
3         const errors = response.getElementsByTagName("error");
4         if (errors.length != 0) {
5             // TODO: update these when the v htmls are done
6             createStatusElements(errors, "form");
7         } else {
8             alert("Cliente creado correctamente");
9             window.location = "./index_operarios.html";
10         }
11     });
12 }

```

```

1 function createAccount() {
2     // TODO: accounts are only created with these now
3     fetch2(["user", "password", "dni"], [], "registro_cuentas.pl", function(response) {
4         if (errors.length != 0) {
5             // TODO: update these when the v htmls are done
6             createStatusElements(errors, "form");
7         } else {
8             alert("Cuenta creada correctamente");
9             window.location = "./index_operarios.html";
10         }
11     });
12 }

```

```

1 function createCard() {
2     // TODO: cards are only created with these now
3     fetch2(["number", "password", ["currency", "soles", "dolares"], "dni"], [], "registro_tarjetas.pl",
4         function(response) {
5             if (errors.length != 0) {
6                 // TODO: update these when the v htmls are done
7                 createStatusElements(errors, "form");
8             } else {
9                 alert("Cuenta creada correctamente");
10                 location = "./index_operarios.html";
11             }
12         });
13 }

```

```

14 function generateNumber() {
15     let number = "";
16     for (let i = 0; i < 16; i++) {
17         number += Math.floor(Math.random() * 9);
18     }
19     document.getElementById("number").value = number;
20 }
21

```

```

22 function generatePassword() {
23     const chars = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
24     let password = "";
25     for (let i = 0; i < 10; i++) {
26         password += chars.charAt(Math.random() * chars.length)
27     }
28     document.getElementById("password").value = password;
29 }

```

```

1 function redirect(url) {
2     location = url;
3 }
4
5 // html_params son los parametros que se buscan en el formulario (el id tiene que tener el mismo nombre)
6 // se permite el ingreso de seleccion multiple con un arreglo, el primer elemento es el nombre del parametro y los
7 // siguientes las opciones a buscar en el formulario
8 // extra_params son parametros adicionales que puedes ingresar manualmente "type=operario"
9 // script es la ubicacion del archivo perl, tiene que terminar en .pl
10 // func siempre debe tener el parametro (response)
11 function fetch2(html_params, extra_params, script, func) {
12     let xhttp = new XMLHttpRequest();
13     let query = "";
14     for (let i = 0; i < html_params.length; i++) {
15         try {
16             if (typeof html_params[i] == "object") {
17                 let chosenInput;
18                 for (let j = 1; j < html_params[i].length; j++) {
19                     let input = document.getElementById(html_params[i][j]);
20                     if (input.checked) {
21                         chosenInput = input;
22                         break;
23                     }
24                 }
25                 query += html_params[i][0] + "=" + chosenInput.value + "&";
26             } else {
27                 let param_value = document.getElementById(html_params[i]).value;
28                 query += html_params[i] + "=" + param_value + "&";
29             }
30         } catch (err) {
31             alert("Error obteniendo " + html_params[i]);
32         }
33     }
34     for (let i = 0; i < extra_params.length; i++) {
35         query += extra_params[i] + "&";
36     }
37
38     xhttp.onreadystatechange = function () {
39         if (xhttp.readyState == 4 && xhttp.status == 200) {
40             let response = xhttp.responseXML || new Document();
41             func(response);
42         }
43     };
44
45     xhttp.open("POST", "./cgi-bin/" + script, true);
46     xhttp.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded; charset=ISO-8859-1');
47     xhttp.send(query);
48 }
49
50 function removeStatusElements() {
51     let errorElements = document.getElementsByName("error");
52     while (errorElements.length > 0) {
53         errorElements[0].remove();

```

```

54     }
55 }
56
57 function createStatusElements(errors, parent) {
58     removeStatusElements(parent);
59     for (let i = 0; i < errors.length; i++) {
60         let element = errors[i].getElementsByTagName("element")[0].firstChild.nodeValue;
61         let message = errors[i].getElementsByTagName("message")[0].firstChild.nodeValue;
62         createElement(element, message, parent);
63     }
64 }
65
66 function createElement(element, message, parent) {
67     const statusElement = document.createElement("h3");
68     const placeElement = document.getElementById(element);
69
70     placeElement.classList.add("error");
71     statusElement.classList.add("error");
72
73     statusElement.innerHTML = message;
74     statusElement.setAttribute("name", "error");
75     document.getElementById(parent).insertBefore(statusElement, placeElement);
76 }

```
