

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_names = (1 => "DEPOSITO", 2 => "RETIRO", 3 => "TRANSFERENCIA ENVIADA", 4 => "TRANSFERENCIA RECIBIDA");
25
26 my $cgi = CGI->new;
27 $cgi->charset("UTF-8");
28
29 my %cookies = CGI::Cookie->fetch();
30 my $session_cookie = $cookies{"session_id_cliente"};
31
32 if ($session_cookie) {
33     my $session_id = $session_cookie->value();
34     my $session = CGI::Session->load($session_id);
35     my $card_id = $cgi->param("card_id");
36
37     my $u = "query";
38     my $p = "YR4AFJUC3nyRmasY";
39     my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
40     my $dbh = DBI->connect($dsn, $u, $p);
41
42     my $sth = $dbh->prepare("SELECT 'tarjetas'.moneda, 'movimientos'.monto, 'movimientos'.tipo,
43                             'movimientos'.fecha
44                             FROM tarjetas, movimientos
45                             WHERE tarjetas.id = '$card_id' AND movimientos.tarjeta_id = tarjetas.id");
46
47     $sth->execute;
48
49     my @row = $sth->fetchrow_array;
50     my $currency = $row[0] eq "S/" ? "S/." : "\$";
51     my $amount = $currency.$row[1];
52     my $type = $type_names{$row[2]};
53     my $date = $row[3];
54     my $total = $row[1] * get_mult($row[2]);
55
56     print $cgi->header("text/xml");
57     print "<status>\n";
58     print_row($amount, $type, $date);
59     while (@row = $sth->fetchrow_array) {
```

```

58     $total = $total + ($row[1] * get_mult($row[2]));
59     my $amount = $currency.$row[1];
60     my $type = $type_names{$row[2]};
61     my $date = $row[3];
62     print_row($amount, $type, $date);
63 }
64 my $balance = $currency.$total;
65 print<<XML;
66     <balance>$balance</balance>
67 </status>
68 XML
69 }
70
71 sub get_mult {
72     my $mult = 1;
73     my $type_code = $_[0];
74     if ($type_code == 2 || $type_code == 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
45     $sth->execute;
46
47     print $cgi->header("text/xml");
48     print "<cards>\n";
49     while (my @row = $sth->fetchrow_array) {
50         my $id = $row[0];
51         my $card_number = $row[1];
52         my $code = $row[2];
53         my $expire_date = $row[3];
54         my $currency = $row[4] eq "s" ? "SOLES" : "DÓLARES";
55
56         print_card($id, $card_number, $code, $expire_date, $currency);
57     }
58     print "</cards>\n";
59 }
60
61 sub print_card {
62     print<<XML;
63     <card>
64         <id>$_[0]</id>
65         <card_number>$_[1]</card_number>
66         <code>$_[2]</code>
67         <expire_date>$_[3]</expire_date>
68         <currency>$_[4]</currency>
69     </card>
70     XML
71 }

```

```

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, t), amount, other_card (only for t)
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, t => 3);
25
26 my $cgi = CGI->new;
27 $cgi->charset("UTF-8");
28 my $card_id = $cgi->param("card_id");
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 our $other_id;
44
45 my %errors = (card_id => $card_id_status, type => $type_status, amount => $amount_status);
46
47 if ($session_cookie) {
48     transaction();
49 }
50
51 sub transaction {
52     print $cgi->header("text/xml");
53     if (!$card_id_status && !$amount_status && !$type_status) {
54         my $type_num = $type_nums{$type};
55         if ($type_num == 3) {
56             transference();
57         }
58         my $sth = $dbh->prepare("INSERT INTO movimientos (tarjeta_id, monto, tipo) VALUES ('$card_id', '$amount',
59             '$type_num')");
60         $sth->execute;
61         return;
62     }
63     print_errors();
64 }
65
66
67 sub transference {
68     my $other_card = $cgi->param("other_card");
69     my $other_card_status = check_other_card($other_card);
70     if (!$other_card_status) {
71         my $sth = $dbh->prepare("INSERT INTO movimientos (tarjeta_id, monto, tipo) VALUES ('$other_id', '$amount',
72             '4')");
73         $sth->execute;
74         return;
75     }
76     print<<XML;
77     <error>
78         <element>other_card</element>
79         <message>$other_card_status</message>
80     </error>

```

```

81 XML
82 }
83
84 sub check_card_id {
85     my $card_id = $_[0];
86     my $length = length($card_id);
87     if (!$card_id || $length == 0) {
88         return "Tarjeta no valida.";
89     }
90 }
91
92 sub check_amount {
93     my $amount = $_[0];
94     if (!$amount) {
95         return "Ingrese una cantidad.";
96     }
97     if ($amount !~ /\d+$/ || $amount < 0) {
98         return "Cantidad no valida.";
99     }
100     if ($type eq "w") {
101         my $sth = $dbh->prepare("SELECT SUM(monto*tipo) FROM movimientos WHERE tarjeta_id = '$card_id'");
102         $sth->execute;
103         my @sum = $sth->fetchrow_array;
104         my $balance = $sum[0] || 0;
105         if ($balance - $amount < 0) {
106             return "Balance negativo."
107         }
108     }
109 }
110
111 sub check_type {
112     my $type = $_[0];
113     if (!$type) {
114         return "Marque un tipo";
115     }
116     if ($type !~ /^[dwt]$/) {
117         return "Tipo no valido";
118     }
119 }
120
121 sub check_other_card {
122     my $number = $_[0];
123     if (!$number) {
124         return "Ingrese un número de tarjeta.";
125     }
126     if ($number !~ /\d{16}$/) {
127         return "Número de tarjeta no valido.";
128     }
129     my $sth = $dbh->prepare("SELECT 'id' FROM tarjetas WHERE numero = '$number'");
130     $sth->execute;
131     my @row = $sth->fetchrow_array;
132     if (!@row) {
133         return "La tarjeta no existe.";
134     }
135     $other_id = $row[0];
136     return undef;
137 }
138
139 sub print_errors {
140     print "<errors>\n";
141     for my $key (keys %errors) {
142         if ($errors{$key}) {
143             print<<XML;
144             <error>

```

```

145     <element>$key</element>
146     <message>$errors{$key}</message>
147 </error>
148 XML
149     }
150 }
151 print "</errors>\n";
152 }

```

```

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, plastname => $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 !~ /^[a-zA-Z]{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 ($plastname !~ /^[a-zA-Z]{1,30}$/) {
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 ($mlastname !~ /^[a-zA-Z]{1,30}$/) {
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 my $u = "query";
30 my $p = "YR4AFJUC3nyRmasY";
31 my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
32 my $dbh = DBI->connect($dsn, $u, $p);
33
34 our $client_id;
35 my $user_status = check_user($user);
36 my $password_status = check_password($password);
37 my $dni_status = check_DNI($dni);
38
39 my %errors = (user => $user_status, password => $password_status, dni => $dni_status);
40
41 register();
42
43 sub register {
44     if (!$user_status && !$password_status && !$dni_status) {
45         my $sth = $dbh->prepare("INSERT INTO cuentas (usuario, clave, cliente_id) VALUES ('$user', '$password',

```

```

        '$client_id');"');
46     $sth->execute;
47     print $cgi->header("text/xml");
48
49     return;
50 }
51
52 print_errors();
53 }
54
55 sub check_user {
56     my $card_number = $_[0];
57     my $length = length($card_number);
58     if (!$card_number || $length == 0) {
59         return "Ingreso un usuario.";
60     }
61     if ($length > 30) {
62         return "Usuario muy largo.";
63     }
64 }
65
66 sub check_password {
67     my $password = $_[0];
68     my $length = length($password);
69     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     return undef;
93 }
94
95 sub print_errors {
96     print $cgi->header("text/xml");
97     print "<errors>\n";
98     for my $key (keys %errors) {
99         if ($errors{$key}) {
100             print<<XML;
101             <error>
102                 <element>$key</element>
103                 <message>$errors{$key}</message>
104             </error>
105             XML
106         }
107     }
108     print "</errors>\n";

```

```
109 }

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 my $u = "query";
34 my $p = "YR4AFJUC3nyRmasY";
35 my $dsn = "dbi:mysql:database=bancafinal;host=127.0.0.1";
36 my $dbh = DBI->connect($dsn, $u, $p);
37
38 our $account_id;
39 my $number_status = check_number($number);
40 my $password_status = check_password($password);
41 my $currency_status = check_currency($currency);
42 my $dni_status = check_DNI($dni);
43
44 my %errors = (number => $number_status, password => $password_status, currency => $currency_status, dni =>
    $dni_status);
45
46 register();
47
48 sub register {
49     if (!$number_status && !$password_status && !$currency_status && !$dni_status) {
50         my $sth = $dbh->prepare("INSERT INTO tarjetas (numero, clave, vencimiento, cuenta_id) VALUES ('$number',
            '$password', '$expire_date', '$account_id')");
51         $sth->execute();
52         print $cgi->header("text/xml");
53
54         return;
55     }
56
57     print_errors();
58 }
59
```

```

60 sub check_number {
61     my $number = $_[0];
62     if (!$number) {
63         return "Ingrese un número de tarjeta.";
64     }
65     if ($number !~ /\d{16}$/) {
66         return "Número de tarjeta no valido.";
67     }
68 }
69
70 sub check_password {
71     my $password = $_[0];
72     if (!$password) {
73         return "Ingrese una clave.";
74     }
75     if (length($password) > 30) {
76         return "Clave muy larga.";
77     }
78     if (length($password) < 8) {
79         return "Clave muy corta.";
80     }
81 }
82
83 sub check_currency {
84     my $currency = $_[0];
85     if (!$currency) {
86         return "Marque una moneda.";
87     }
88     if ($currency !~ /^[sd]$/) {
89         return "Moneda no valida.";
90     }
91 }
92
93 sub check_DNI {
94     my $dni = $_[0];
95     if (!$dni) {
96         return "Ingrese un DNI.";
97     }
98     if ($dni !~ /\d{8}$/) {
99         return "DNI no valido.";
100     }
101     my $sth = $dbh->prepare("SELECT 'cuentas'. 'id'
102                             FROM cuentas, clientes
103                             WHERE clientes.dni = '$dni' AND cuentas.cliente_id = clientes.id");
104     $sth->execute;
105     my @row = $sth->fetchrow_array;
106     if (!@row) {
107         return "Cliente no existente."
108     }
109     $account_id = $row[0];
110     return undef;
111 }
112
113 sub print_errors {
114     print $cgi->header("text/xml");
115     print "<errors>\n";
116     for my $key (keys %errors) {
117         if ($errors{$key}) {
118             print<<XML;
119             <error>
120                 <element>$key</element>
121                 <message>$errors{$key}</message>
122             </error>
123     XML

```

```

124     }
125 }
126 print "</errors>\n";
127 }

```

```

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.js"></script>
11  <script src="./scripts/estado_tarjeta.js"></script>
12  <script src="./scripts/lista_tarjetas.js"></script>
13  <script src="./scripts/slide.js"></script>
14 </head>
15
16 <body onload="init('cliente')">
17   <div id="content" class="content">
18     <div class="top">
19       <h2>Estado de cuenta</h2>
20       <div class="righTop">
21         <a href="./index_clientes.html" class="btnLink">Volver</a>
22         <div>
23           <h5 id="timer">Su sesión concluirá en:</h5>
24           <input type="button" class="logout" onclick="closeSession('cliente')" value="Cerrar sesión" />
25         </div>
26       </div>
27     </div>
28     <div class="cards">
29       <h3 class="label-slide">Selecciona una tarjeta</h3>
30       
31       <div id="slide" class="slide">
32
33       </div>
34       
35     </div>
36     <input type="button" value="Obtener estado" class="submitButton" onclick="getStatus()" />
37     <div class="form">
38       <h3>Balance de cuenta:</h3>
39       <div id="balance"></div>
40       <h3>Últimos movimientos:</h3>
41     </div>
42     <div id="movements" style="width: 100%;">
43       <div class="movement">
44         <h5 style="font-size: 20px;">CANTIDAD</h5>
45         <h5 style="font-size: 20px;">TIPO</h5>
46         <h5 style="font-size: 20px;">FECHA</h5>
47       </div>
48     </div>
49   </div>
50 </body>
51
52 </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.js"></script>
10     <script src="./scripts/utilidades.js"></script>
11 </head>
12
13 <body onload="init('cliente')">
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             <h5 id="timer">Su sesión concluirá en:</h5>
25             <input type="button" class="logout" onclick="closeSession('cliente')" value="Cerrar sesión" />
26         </div>
27     </div>
28 </div>
29 </body>
30 </html>
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.js"></script>
11 </head>
12
13 <body onload="init('operario')">
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                 <h5 id="timer">Su sesión concluirá en:</h5>
22                 <input type="button" class="logout" onclick="closeSession('operario')" 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 </html>
33
34 </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.css" />
9   <script src="./scripts/utilidades.js"></script>
10  <script src="./scripts/login.js"></script>
11 </head>
12
13 <body>
14   <div id="content" class="content">
15     
16   </div>
17   <h1>Ingreso de Clientes</h1>
18   <div id="form" class="form">
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('cliente')" value="Ingresar"/>
22     <a href="login_operarios.html">Soy un operario</a>
23   </div>
24 </body>
25
26 </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.css" />
9   <script src="./scripts/utilidades.js"></script>
10  <script src="./scripts/login.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('operario')" value="Ingresar" />
21       <a href="login_clientes.html">Soy un cliente</a>
22     </div>
23   </div>
24 </body>
25
26 </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.js"></script>

```

```

11 <script src="./scripts/movimiento_cliente.js"></script>
12 <script src="./scripts/lista_tarjetas.js"></script>
13 <script src="./scripts/slide.js"></script>
14 </head>
15
16 <body onload="init('cliente')">
17   <div id="content" class="content">
18     <div class="top">
19       <h2>
20         Asistente de movimientos
21       </h2>
22       <div id="welcome"></div>
23       <div class="righthTop">
24         <a href="./index_clientes.html" class="btnLink">Volver</a>
25       </div>
26       <h5 id="timer">Su sesión concluirá en:</h5>
27       <input type="button" class="logout" onclick="closeSession('cliente')" value="Cerrar sesión" />
28     </div>
29   </div>
30
31   <div id="form" class="form">
32     <div class="cards">
33       <h3 class="label-slide">Selecciona una tarjeta</h3>
34       
35       <div id="slide" class="slide">
36
37
38     </div>
39     
40   </div>
41   <h3>Tipo de movimiento:</h3>
42   <div id="type" class="options">
43     <input type="radio" id="deposit" name="type" value="d" onclick="setCardInput(false)" checked />
44     <label for="d">Depósito</label>
45     <input type="radio" id="withdrawal" name="type" value="w" onclick="setCardInput(false)" />
46     <label for="w">Retiro</label>
47     <input type="radio" id="transference" name="type" value="t" onclick="setCardInput(true)" />
48     <label for="t">Transferencia</label>
49   </div>
50   <input type="text" id="amount" placeholder="Cantidad" class="inputText" />
51   <input type="text" style="display: none;" id="other_card" placeholder="Otra tarjeta" class="inputText" />
52   <input type="button" class="submitButton" onclick="doMovement()" value="Realizar movimiento" />
53 </div>
54 </div>
55 </body>
56
57 </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.js"></script>
11  <script src="./scripts/registro_clientes.js"></script>
12 </head>
13
14 <body onload="init('operario')">
15   <div id="content" class="content">

```

```

16     <div class="leftTop">
17         <div id="welcome"></div>
18     </div>
19     <div class="righthTop">
20         <a href="./index_operarios.html" class="btnLink">Volver</a>
21         <div>
22             <h5 id="timer">Su sesión concluirá en:</h5>
23             <input type="button" class="logout" onclick="closeSession('operario')" value="Cerrar sesión" />
24         </div>
25     </div>
26 </div>
27 <h2>Asistente de creación de clientes</h2>
28 <div id="form" class="form">
29     <input type="text" id="dni" class="inputText" placeholder="DNI" />
30     <input type="text" id="name" class="inputText" placeholder="Nombres completos" />
31     <input type="text" id="plastname" class="inputText" placeholder="Apellido paterno" />
32     <input type="text" id="mlastname" class="inputText" placeholder="Apellido materno" />
33     <input type="date" id="bdate" class="inputText" placeholder="Fecha de nacimiento" />
34     <input type="button" class="submitButton" onclick="createClient()" value="Crear clientes" />
35 </div>
36 </body>
37
38 </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.js"></script>
11    <script src="./scripts/registro_cuentas.js"></script>
12 </head>
13
14 <body onload="init('operario')">
15     <div id="content" class="content">
16         <div class="leftTop">
17             <div id="welcome"></div>
18         </div>
19         <div class="righthTop">
20             <a href="./index_operarios.html" class="btnLink">Volver</a>
21             <div>
22                 <h5 id="timer">Su sesión concluirá en:</h5>
23                 <input type="button" class="logout" onclick="closeSession('operario')" value="Cerrar sesión" />
24             </div>
25         </div>
26     </div>
27     <h2>Asistente de creación de cuentas</h2>
28     <div id="form" class="form">
29         <input type="text" class="inputText" id="user" placeholder="Usuario" />
30         <input type="password" class="inputText" id="password" placeholder="Contraseña" />
31         <input type="text" class="inputText" id="password_output" placeholder="Contraseña generada" readonly="true" />
32         <input type="button" onclick="generatePassword()" value="Generar contraseña" class="submitButton" />
33         <input type="text" id="dni" placeholder="DNI del cliente" class="inputText" />
34         <input type="button" class="submitButton" onclick="createAccount()" value="Crear cuenta" />
35     </div>
36 </body>
37
38 </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.js"></script>
11    <script src="./scripts/registro_tarjetas.js"></script>
12 </head>
13
14 <body onload="init('operario')">
15     <div id="content" class="content">
16         <div class="leftTop">
17             <div id="welcome"></div>
18         </div>
19         <div class="righthTop">
20             <a href="./index_operarios.html" class="btnLink">Volver</a>
21             <div>
22                 <h5 id="timer">Su sesión concluirá en:</h5>
23                 <input type="button" class="logout" onclick="closeSession('operario')" value="Cerrar sesión" />
24             </div>
25         </div>
26     </div>
27     <h2>Asistente de creación de tarjetas</h2>
28     <div id="form" class="form">
29         <div>
30             <input type="text" class="inputText" id="number" placeholder="Número de tarjeta" />
31             <input type="button" class="submitButton" class="submit" onclick="generateNumber()" value="Generar número"
32             />
33         </div>
34         <div>
35             <input type="password" class="inputText" id="password" placeholder="Contraseña" />
36             <input type="text" class="inputText" id="password_output" placeholder="Contraseña generada" />
37             <input type="button" class="submitButton" class="submit" onclick="generatePassword()" value="Generar
38             clave" />
39         </div>
40         <div>
41             <h3>Tipo de moneda:</h3>
42             <div id="currency" class="options">
43                 <label for="soles">
44                     <input type="radio" id="soles" name="currency" value="s" checked />
45                     Soles
46                 </label>
47                 <label for="dollars">
48                     <input type="radio" id="dollars" name="currency" value="d" />
49                     Dólares
50                 </label>
51             </div>
52         </div>
53         <input class="inputText" type="text" id="dni" placeholder="DNI del cliente" />
54         <input class="submitButton" type="button" class="submit" onclick="createCard()" value="Crear tarjeta" />
55     </div>
56 </body>
57 </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     text-align: center;
43 }
44
45 /* Messages */
46 h3 {
47     text-align: center;
48     font-size: 20px;
49     color: #002a8d;
50 }
51
52 .error {
53     color: #e30425;
54 }
55
56 /* Text */
57 h4 {
58     text-align: center;
59     font-size: 15px;
60     color: #202e44;
61 }
```

```

62
63 /* Text - Inline */
64 h5 {
65     flex: 1;
66     display: inline-block;
67     text-align: center;
68     font-size: 15px;
69     color: #202e44;
70 }
71
72 h6 {
73     color: white;
74     margin: 0px;
75     padding: 0px;
76 }
77
78 /* Links */
79 a {
80     text-align: left;
81     font-size: 20px;
82     text-decoration: none;
83     color: #202e44;
84 }
85
86 a:hover {
87     text-decoration: underline;
88 }
89
90 .content {
91     display: flex;
92     flex-direction: column;
93     justify-content: start;
94     align-items: center;
95 }
96
97 .btnLink {
98     color: #202E44;
99     text-align: center;
100    font-size: 16px;
101    font-style: normal;
102    font-weight: normal;
103    line-height: normal;
104    display: flex;
105    align-items: center;
106    border-radius: 20px;
107    border: 1px solid #ACB2BD;
108    padding: 10px 16px;
109    width: max-content;
110    margin-top: 6px;
111    transition: box-shadow .1s ease-out;
112    width: 100%;
113    display: flex;
114    justify-content: center;
115    align-items: center;
116    min-width: max-content;
117 }
118
119 .btnLink:hover {
120     text-decoration: none;
121     box-shadow: 00px 0px 5px gray;
122 }
123
124 .righthTop {
125     display: flex;

```

```

126     justify-content: center;
127     align-items: center;
128     gap: 20px;
129 }
130
131 @media screen and (max-width: 700px) {
132     .righthTop {
133         flex-direction: column;
134     }
135 }
136
137 .logout {
138     width: 150px;
139     font-size: 15px;
140     padding: 5px;
141     display: block;
142     border: none;
143     border-radius: 10px;
144     background-color: #ff7800;
145     color: white;
146 }
147
148 .logout:hover {
149     background-color: #ff961f;
150     box-shadow: 0 4px 0px #c8c8c8;
151     cursor: pointer;
152 }
153
154 .logout:active {
155     background-color: #e45e00;
156 }

```

```

1  * {
2     box-sizing: border-box;
3     overflow: hidden;
4 }
5
6 body {
7     margin-top: 8svh;
8     display: flex;
9     flex-direction: column;
10    justify-content: start;
11    align-items: center;
12 }
13
14 h1 {
15     font-size: 2.5rem;
16 }
17
18 .logo {
19     width: 125px;
20 }
21
22 .logo:hover {
23     cursor: pointer;
24 }
25
26 .content {
27     width: 100%;
28     max-width: 1200px;
29     display: flex;
30     justify-content: start;
31     align-items: center;

```

```

32 }
33
34 .inputText::placeholder {
35   color: #99a1ad;
36 }
37
38 .inputText: hover::placeholder {
39   color: #606c7f;
40 }
41
42 .inputText:focus {
43   outline: none;
44   border-width: 2px;
45 }
46
47 .inputText {
48   display: block;
49   margin: auto;
50   padding: 15px;
51   border: 0;
52   border-bottom: solid 1px gray;
53   width: 100%;
54   font-size: 1.25rem;
55   color: #202e44;
56   max-width: 400px;
57   background-color: #ffff;
58 }
59
60 input[type="text"][class="error"],
61 input[type="password"][class="error"] {
62   border-color: #e30425;
63 }
64
65 /* Button Styles */
66 .submitButton {
67   display: block;
68   border: none;
69   border-radius: 10px;
70   background-color: #ff7800;
71   color: white;
72   margin: auto;
73   max-width: 400px;
74   width: 100%;
75   font-size: 1.2rem;
76   padding: 15px 20px;
77   margin-top: 30px;
78   font-weight: 600;
79 }
80
81 .submitButton: hover {
82   background-color: #ff961f;
83   box-shadow: 0 4px 4px 0px #c8c8c8;
84   cursor: pointer;
85 }
86
87 .submitButton: active {
88   background-color: #e45e00;
89 }
90
91 .form {
92   box-sizing: border-box;
93   display: flex;
94   flex-direction: column;
95   gap: 20px;

```

```

96     width: 80%;
97     max-width: 400px;
98     background: #ffffff;
99     border-radius: 10px;
100    padding: 30px;
101 }
102
103 @media screen and (max-width: 400px) {
104     :root {
105         font-size: 14px;
106     }
107     body {
108         margin: 0;
109         padding-bottom: 10vh;
110         justify-content: center;
111         align-items: center;
112     }
113 }

```

```

1  * {
2      box-sizing: border-box;
3      overflow: hidden;
4  }
5
6  body{
7      margin-top: 8svh;
8  }
9  h1 {
10     font-size: 2.5rem;
11 }
12
13 /* Logo Styles */
14 .logo {
15     width: 125px;
16 }
17
18 .logo:hover {
19     cursor: pointer;
20 }
21
22 /* Input Styles */
23 .inputText::placeholder{
24     color: #99a1ad;
25 }
26
27 .inputText:hover::placeholder {
28     color: #606c7f;
29 }
30
31
32 .inputText:focus {
33     outline: none;
34     border-width: 2px;
35 }
36
37 input[type="text"][class=error],
38 input[type="password"][class=error] {
39     border-color: #e30425;
40 }
41
42 .submitButton:hover {
43     background-color: #ff961f;
44     box-shadow: 0 4px 0px #c8c8c8;

```

```

45     cursor: pointer;
46 }
47
48 .submitButton:active {
49     background-color: #e45e00;
50 }
51
52 .form {
53     box-sizing: border-box;
54     display: flex;
55     flex-direction: column;
56     gap: 20px;
57     width: 100%;
58     max-width: 400px;
59     background: #ffffff;
60     border-radius: 10px;
61     padding: 30px;
62 }
63
64 .inputText {
65     display: block;
66     margin: auto;
67     padding: 15px;
68     border: 0px;
69     border-bottom: solid 1px gray;
70     width: 100%;
71     font-size: 20px;
72     color: #202e44;
73     max-width: 400px;
74     background-color: transparent;
75 }
76
77 .submitButton {
78     display: block;
79     border: none;
80     border-radius: 10px;
81     background-color: #ff7800;
82     color: white;
83     margin: auto;
84     max-width: 400px;
85     width: 100%;
86     font-size: 1.2rem;
87     padding: 15px 20px;
88     margin-top: 30px;
89 }

```

```

1  /* Input Styles */
2  * {
3      box-sizing: border-box;
4  }
5
6  body {
7      display: flex;
8      flex-direction: column;
9      justify-content: start;
10     align-items: center;
11 }
12
13 .righthTop {
14     display: flex;
15     justify-content: center;
16     align-items: center;
17     gap: 20px;

```

```

18 }
19
20 @media screen and (max-width: 700px) {
21   .righthTop {
22     flex-direction: column;
23   }
24 }
25
26 .content {
27   width: 100%;
28   max-width: 1200px;
29   display: flex;
30   justify-content: space-between;
31   align-items: center;
32   flex-direction: column;
33   padding: 0px 20px;
34 }
35
36 .options {
37   display: flex;
38   margin: auto;
39   justify-content: center;
40 }
41
42 /* Button Styles */
43 .submitButton {
44   margin: auto;
45   display: block;
46   border: none;
47   border-radius: 25px;
48   background-color: #ff7800;
49   color: white;
50 }
51
52 .submitButton:hover {
53   background-color: #ff961f;
54   box-shadow: 0 4px 0px #c8c8c8;
55   cursor: pointer;
56 }
57
58 .submitButton:active {
59   background-color: #e45e00;
60 }
61
62 .submit {
63   width: 225px;
64   font-size: 20px;
65   padding: 5px;
66 }
67
68 /* Division Styles */
69 .movement {
70   display: flex;
71   justify-content: space-between;
72   width: 100%;
73   max-width: 1400px;
74 }
75
76 .form {
77   width: 100%;
78   max-width: 600px;
79   display: flex;
80   flex-direction: column;
81   justify-content: start;

```

```

82     align-items: center;
83     gap: 20px;
84 }
85
86 .submitButton {
87     display: block;
88     border: none;
89     border-radius: 10px;
90     background-color: #ff7800;
91     color: white;
92     margin: auto;
93     max-width: 400px;
94     width: 100%;
95     font-size: 1.2rem;
96     padding: 15px 20px;
97 }
98
99 .inputText {
100     display: block;
101     margin: auto;
102     padding: 15px;
103     border-style: solid;
104     border-radius: 10px;
105     border-width: 1px;
106     border-color: #868f9e;
107     width: 100%;
108     font-size: 20px;
109     color: #202e44;
110 }
111
112 /* Estilos para el slide de tarjetas */
113
114 .cards {
115     width: 500px;
116     display: grid;
117     grid-template-areas:
118         "label label label"
119         "prev slide next";
120     align-items: center;
121     margin-bottom: 32px;
122     gap: 12px;
123 }
124
125 .control {
126     height: 25px;
127     transition: transform 0.3s ease;
128 }
129
130 .control:hover {
131     cursor: pointer;
132     transform: scale(1.2);
133 }
134
135 .label-slide {
136     grid-area: label;
137     display: flex;
138     justify-content: center;
139     align-items: center;
140 }
141
142 .next {
143     grid-area: next;
144 }
145

```



```

146 .prev {
147     grid-area: prev;
148 }
149
150 .slide {
151     display: flex;
152     overflow: hidden;
153     width: 400px;
154     box-shadow: 0 0 15px 7px #0005;
155     height: 238px;
156     border-radius: 10px;
157     scroll-behavior: smooth;
158 }
159
160 .slide::-webkit-scrollbar {
161     display: none;
162 }
163
164 .card {
165     display: flex;
166     min-width: 400px;
167     height: 238px;
168     padding-inline: 30px;
169     padding-block: 10px;
170     border-width: 1px;
171     background: url(https://pc-solucion.es/wp-content/uploads/2019/02/como-hacer-un-degradado-css.jpg);
172     color: #fff;
173     flex-direction: column;
174     justify-content: space-between;
175 }
176
177 h4 {
178     margin: 0;
179     color: #fff;
180     font-size: 1.1rem;
181     font-weight: 600;
182 }
183
184 .info>label {
185     font-size: .9rem;
186 }
187
188 .info {
189     display: flex;
190     flex-direction: column;
191     align-items: start;
192     gap: 10px;
193     margin-block: 5px;
194 }
195
196 .info.exp {
197     align-self: flex-end;
198 }
199
200 .card-selector {
201     opacity: 0;
202     height: 0;
203 }

```

```

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 h1 {
13     font-size: 2.5rem;
14 }
15
16 .logo {
17     width: 125px;
18 }
19
20 .logo:hover {
21     cursor: pointer;
22 }
23
24 .content {
25     width: 100%;
26     max-width: 1200px;
27     display: flex;
28     flex-direction: row;
29     justify-content: space-between;
30     align-items: center;
31 }
32
33 .inputText::placeholder {
34     color: #99a1ad;
35 }
36
37 .inputText:hover::placeholder {
38     color: #606c7f;
39 }
40
41 .inputText:focus {
42     outline: none;
43     border-width: 2px;
44 }
45
46 .inputText {
47     display: block;
48     margin: auto;
49     padding: 15px;
50     border: 0;
51     border-bottom: solid 1px gray;
52     width: 100%;
53     font-size: 1.25rem;
54     color: #202e44;
55     max-width: 400px;
56     background-color: #ffff;
57 }
58
59 input[type="text"][class="error"],
60 input[type="password"][class="error"] {
61     border-color: #e30425;
62 }
63
64 /* Button Styles */
65 .submitButton {
66     display: block;
67     border: none;
68     border-radius: 10px;

```

```

69     background-color: #ff7800;
70     color: white;
71     margin: auto;
72     max-width: 400px;
73     width: 100%;
74     font-size: 1.2rem;
75     padding: 15px 20px;
76     margin-top: 30px;
77     font-weight: 600;
78     transition: all 200ms;
79 }
80
81 .submitButton:hover {
82     background-color: #ff961f;
83     box-shadow: 0 4px 4px 0px #c8c8c8;
84     cursor: pointer;
85 }
86
87 .submitButton:active {
88     background-color: #e45e00;
89 }
90
91 .form {
92     box-sizing: border-box;
93     display: flex;
94     flex-direction: column;
95     gap: 20px;
96     width: 80%;
97     max-width: 400px;
98     background: #ffffff;
99     border-radius: 10px;
100    padding: 30px;
101 }
102
103 @media screen and (max-width: 400px) {
104     :root {
105         font-size: 14px;
106     }
107
108     body {
109         margin: 0;
110         padding-bottom: 10vh;
111         justify-content: center;
112         align-items: center;
113     }
114 }
115
116
117 .options {
118     display: flex;
119     justify-content: space-around;
120     align-items: center;
121     font-size: 20px;
122 }

```

5. JavaScript:

```
1 function getStatus() {
2     fetch2([generateCardsQuery()], [], "estado_tarjeta.pl", function (response) {
3         removeExistingElements();
4
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 removeExistingElements() {
19     let balanceElement = document.getElementById("balance_elem");
20     let movementsElements = document.getElementsByName("movements_elem");
21     if (balanceElement) {
22         balanceElement.remove();
23     }
24     // Recuerdos de 'concurrent modification exception'
25     const leng = movementsElements.length;
26     for (let i = 0; i < leng; i++) {
27         movementsElements[i].remove();
28     }
29 }
30
31 function createBalanceElement(balance) {
32     const balanceElement = document.createElement("h5");
33     balanceElement.style = "font-size: 20px;";
34     balanceElement.innerHTML = balance;
35     balanceElement.id = "balance_elem";
36     document.getElementById("balance").appendChild(balanceElement);
37 }
38
39 function createMovementElement(values) {
40     const movementElement = document.createElement("div");
41     movementElement.setAttribute("name", "movements_elem");
42     movementElement.classList.add("movement");
43
44     for (let i = 0; i < values.length; i++) {
45         createMovementSubElement(movementElement, values[i]);
46     }
47
48     document.getElementById("movements").appendChild(movementElement);
49 }
50
51 function createMovementSubElement(parentElement, value) {
52     const element = document.createElement("h5");
53     element.innerHTML = value;
54     parentElement.appendChild(element);
55 }
56
57 addEventListener("load", getCards);
58
59 function getCards() {
60     fetch2([], [], "lista_tarjetas.pl", function (response) {
```

```

5     const cards = response.getElementsByTagName("card");
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        createCardElement([id, card_number, code, currency, expire_date]);
13    }
14    initCardSlide();
15    updateSelectedCard();
16    });
17 }
18
19 function createCardElement(values) {
20     const cardElement = document.createElement("label");
21     cardElement.className = "card mockup";
22
23     const idElement = document.createElement("input");
24     idElement.type = "radio";
25     idElement.className = "card-selector"
26     idElement.name = "card";
27     idElement.id = values[0];
28     idElement.value = values[0];
29     cardElement.appendChild(idElement);
30
31     let number = formatNumber(values[1]);
32     let password = formatPassword(values[2]);
33     let expire = formatExpire(values[4]);
34     createCardSubElement("info", "Número de cuenta", number, cardElement);
35     createCardSubElement("info", "Clave", password, cardElement);
36     createCardSubElement("info exp", "Moneda", values[3], cardElement);
37     createCardSubElement("info exp", "Fecha Exp", expire, cardElement);
38
39     document.getElementById("slide").appendChild(cardElement);
40 }
41
42 function formatNumber(number) {
43     let fnumber = number.substring(0, 4);
44     for (let i = 1; i <= 3; i++)
45         fnumber += "-" + number.substring(i * 4, i * 4 + 4);
46     return fnumber;
47 }
48
49 function formatPassword(password) {
50     return password.substring(0, 4) + "*".repeat(password.length - 4);
51 }
52
53 function formatExpire(expire) {
54     return expire.substring(0, expire.indexOf(" "));
55 }
56
57 function createCardSubElement(cssclass, title, value, parentElement) {
58     const element = document.createElement("div");
59     element.className = cssclass;
60
61     const titleElement = document.createElement("h6");
62     titleElement.innerHTML = title;
63     const textElement = document.createElement("h4");
64     textElement.innerHTML = value;
65     element.appendChild(titleElement)
66     element.appendChild(textElement);
67
68     parentElement.appendChild(element);

```

```

69 }
70
71 function generateCardsQuery() {
72     const cardElements = document.getElementsByName("card");
73     const cards = ["card_id"];
74     for (let i = 0; i < cardElements.length; i++) {
75         cards[i + 1] = cardElements.item(i).value;
76     }
77     return cards;
78 }

```

```

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

```

```

1 function init(type) {
2     firstLog(type);
3     updateTimer(type);
4     setInterval(function () { updateTimer(type) }, 1 * 1000);
5 }
6
7 function firstLog(type) {
8     fetch2([], ["action=check", 'type=${type}'], "sesion.pl", function (response) {
9         let logged_in = response.getElementsByTagName("logged_in")[0].childNodes[0].nodeValue;
10        if (logged_in == "1") {
11            let welcome = document.createElement("h5");
12            let name = response.getElementsByTagName("name")[0].childNodes[0].nodeValue;
13            welcome.innerHTML = "Bienvenido, " + name + " ";
14            document.getElementById("welcome").append(welcome);
15        } else {
16            redirectBack(type);
17        }
18    });
19 }
20
21 function updateTimer(type) {
22     fetch2([], ["action=check", 'type=${type}'], "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(type);
29         }
30     });
31 }
32
33 function closeSession(type) {
34     fetch2([], ["action=close", 'type=${type}'], "sesion.pl", function (response) {
35         redirectBack(type);
36     })
37 }
38
39 function redirectBack(type) {
40     alert("Su sesión ha expirado");

```

```

41     window.location = './login_${type}s.html';
42 }

```

```

1 function doMovement() {
2     // Delete after changing to forced input
3     let input = document.getElementById("other_card");
4     if (input.style.display == "block")
5         input.value = input.value.replaceAll("-", "");
6     fetch2([generateCardsQuery(), ["type", "deposit", "withdrawal", "transference"], "amount", "other_card"], [],
7         "movimiento_cliente.pl", function (response) {
8         const errors = response.getElementsByTagName("error");
9         if (errors.length != 0) {
10             createStatusElements(errors, "form");
11         } else {
12             alert("Movimiento realizado correctamente");
13             window.location = "./index_clientes.html";
14         }
15     });
16 }

```

```

17 function setCardInput(state) {
18     let input = document.getElementById("other_card");
19     if (state)
20         input.style.display = "block";
21     else
22         input.style.display = "none";
23 }

```

```

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     fetch2(["user", "password", "dni"], [], "registro_cuentas.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("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         const errors = response.getElementsByTagName("error");
6         if (errors.length != 0) {

```

```

6         // TODO: update these when the v htmls are done
7         createStatusElements(errors, "form");
8     } else {
9         alert("Tarjeta creada correctamente");
10        location = "./index_operarios.html";
11    }
12});
13}
14
15function generateNumber() {
16    let number = "";
17    for (let i = 0; i < 16; i++) {
18        number += Math.floor(Math.random() * 9);
19    }
20    document.getElementById("number").value = number;
21}

```

```

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)
11function 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 }
77
78 function generatePassword() {
79     const chars = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
80     let password = "";
81     for (let i = 0; i < 10; i++) {
82         password += chars.charAt(Math.random() * chars.length);
83     }
84     document.getElementById("password").value = password;
85     document.getElementById("password_output").value = password;
86 }

```
