

# Práctica 3: Manual Técnico

Análisis y Diseño de Sistemas 1

## Grupo 5

Integrantes

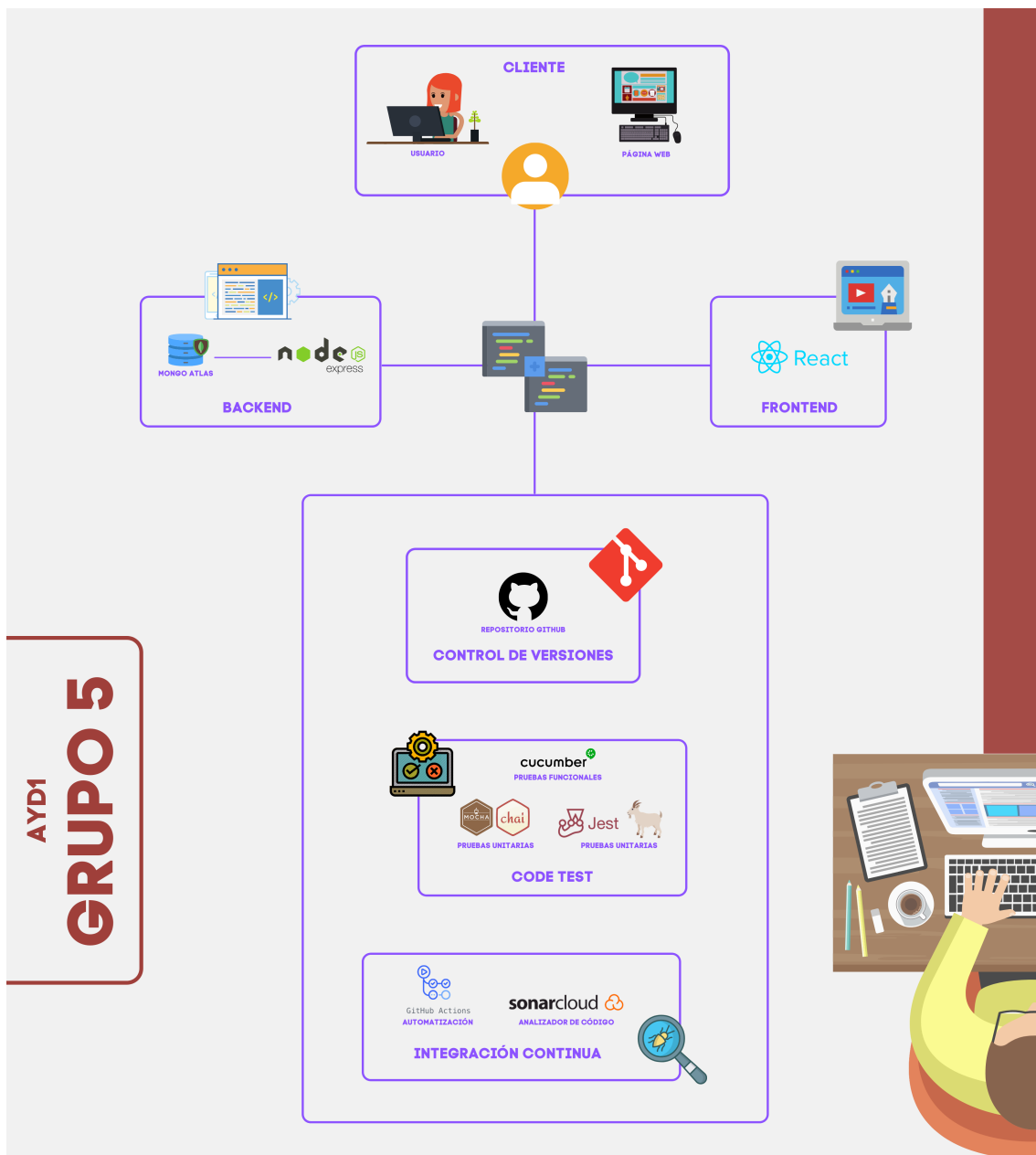
Carné	Nombre
201602625	Oscar Alfredo Llamas Lemus
201504051	Asunción Mariana Sic Sor
201601469	Oscar Eduardo Mazariegos López
201612383	Javier Antonio Álvarez González
201408549	Elba María Álvarez Domínguez

## Tabla de Contenido

- [Diagrama de Arquitectura](#)
- [Casos de Prueba](#)
  - [BackEnd](#)
  - [FrontEnd](#)
- [Anexos](#)
  - [Capturas de pantalla de la aplicación](#)

## Diagrama de Arquitectura

En el siguiente diagrama se observa la arquitectura a utilizar para construir el software requerido.



## Casos de Prueba

### Backend

Caso de Prueba	Prueba Unitaria
<p>¿Qué pasaría si el usuario intenta entrar a la plataforma?</p> <p>¿Qué pasaría si sus credenciales son correctas? Según el método se esperaría un status 202.</p>	<pre>describe('login as a user: ',   ()=&gt;{     it('should login', (done) =&gt; {       chai.request(app).post('/login')</pre>

	<pre>         .send({ cuenta: "37747947969500", password: "123456789"})         .end( function(err,res){             if (err){                 console.log(err);             } else {  console.log(res.body)  expect(res).to.have.status(202);                 done();             }         });     }); }); </pre>
<p>¿Qué pasaría si el usuario desea ingresar a la plataforma?</p> <p>¿Qué pasaría si el usuario provee algún dato erróneo al ingresar a la plataforma? Según el método se esperaría un status 404.</p>	<pre> describe('login as a user: ', ()=&gt;{     it('Shouldn't login', (done) =&gt; {         chai.request(app)         .post('/login')         .send({ cuenta: "37747947969500", password: "123456781"})         .end( function(err,res){             if (err){                 console.log(err);             } else {  console.log(res.body)  expect(res).to.have.status(404);                 done();             }         });     }); }); </pre>
<p>¿Qué pasaría si un usuario que no tiene cuenta desea ingresar a la plataforma?</p> <p>¿Qué pasaría si el CUI no pertenece a ningún otro usuario existente? Según el método se esperaría un status 202.</p>	<pre> describe('Register new user: ', ()=&gt;{     it('Should register a new user', (done) =&gt; {         chai.request(app)         .post('/nuevoUsuario')         .send({             nombre: "Usuario",             apellido: "Prueba",             CUI: </pre>

	<pre> "2448618884917871",     saldo: 175000,     correo: "usuario@gmail.com",     password: "123456781"   })   .end( function(err,res){     if (err){       console.log(err);     } else {  console.log(res.body)  expect(res).to.have.status(202);       done();     }   }); }); }); </pre>
<p>¿Qué pasaría si un usuario que no tiene cuenta desea registrarse en la plataforma?</p> <p>¿Qué pasaría si el CUI con el que desea registrarse ya ha sido utilizado antes? Según el método se esperaría un status 404</p>	<pre> describe('Register new user: ', ()=&gt;{   it(`Shouldn't register a new user`, (done) =&gt; {     chai.request(app)       .post('/nuevoUsuario')       .send({         nombre: "Usuario",         apellido: "Prueba 2",         CUI: "3448688500106",         saldo: 175000,         correo: "usuario2@gmail.com",         password: "123456781"       })       .end( function(err,res){         if (err){           console.log(err);         } else {           console.log(res.body)  expect(res).to.have.status(404);           done();         }       });     });   }); }); </pre>
<p>¿Qué pasaría si se desea obtener la información de cierto usuario?</p>	<pre> describe('Get profile: ',()=&gt;{   it('should get profile', </pre>

<p>¿Qué pasaría si el usuario del cual se solicita información existe? Según el método se esperaría un status 202.</p>	<pre>(done) =&gt; {   chai.request(app)     .post('/login')     .send({ cuenta: "05522194828065", password: "123456789"})     .end( function(err,res){       if (err){         console.log(err);       } else {         console.log(res.body)       }     })     expect(res).to.have.status(202);     done();   }); }); });</pre>
<p>¿Qué pasaría si se desea obtener información de un usuario en particular?</p> <p>¿Qué pasaría si el usuario del que se solicitó la información no está registrado en el sistema? Según el método se esperaría un status 404.</p>	<pre>describe('Get profile: ',()=&gt;{   it('Shouldn't get profile',     (done) =&gt; {       chai.request(app)         .post('/login')         .send({ cuenta: "37747947969500", password: "123456781"})         .end( function(err,res){           if (err){             console.log(err);           } else {             console.log(res.body)           }         })         expect(res).to.have.status(404);         done();       });     }); });</pre>
<p>¿Qué pasaría si se desea obtener el saldo actual de la base de datos de cierto usuario?</p> <p>¿Qué pasaría si el CUI enviado sí pertenece a un usuario ya registrado? Según el método se esperaría un status 202.</p>	<pre>describe('Check balance: ',()=&gt;{   it('Should get user balance',     (done) =&gt; {       chai.request(app)         .post('/consultarSaldo')         .send({ cuenta: "37747947969500"})         .end( function(err,res){           if (err){             console.log(err);           }         })       });     }); });</pre>

	<pre>     } else {       console.log(res.body)        expect(res).to.have.status(202);       done();     }   }); }); }); </pre>
<p>¿Qué pasaría si se desea obtener el saldo actual de la base de datos de cierto usuario?</p> <p>¿Qué pasaría si el CUI enviado no pertenece a un usuario ya registrado? Según el método se esperaría un status 404.</p>	<pre> describe('Check balance: ', ()=&gt;{   it('Shouldn't get user balance', (done) =&gt; {     chai.request(app)       .post('/consultarSaldo')       .send({ cuenta: "3774794796951155"})       .end( function(err,res){         if (err){           console.log(err);         } else {           console.log(res.body)         }          expect(res).to.have.status(404);         done();       }     );   }); }); </pre>
<p>¿Qué pasaría si un usuario desea realizar un nuevo movimiento en su cuenta?</p> <p>¿Qué pasaría si la cuenta de origen si existe registrada en la base de datos? Según el método se esperaría un status 202.</p> <p>¿Qué pasaría si la cuenta a la cual se desea abonar si existe registrada en la base de datos? Según el método se esperaría un status 202.</p>	<pre> describe('New transaction: ', ()=&gt;{   it('Should register a new transaction', (done) =&gt; {     chai.request(app)       .post('/nuevaTransaccion')       .send({  CuentaOrigen:"37747947969500",  CuentaDestino:"05522194828065", monto: 1000, descripcion: "Transaccion exitosa :D" })       .end( function(err,res){         if (err){           console.log(err);         } else {           console.log(res.body)         }       }     );   }); }); </pre>

	<pre> expect(res).to.have.status(202);     done();   } }); }); }); </pre>
<p>¿Qué pasaría si se desea obtener el saldo actual de la base de datos de cierto usuario?</p> <p>¿Qué pasaría si la cuenta de origen y/o la de destino no está registrada en la base de datos? Según el método se esperaría un status 404.</p>	<pre> describe('New transaction: ', ()=&gt;{   it('Shouldn't register a new transaction`, (done) =&gt; {     chai.request(app)       .post('/nuevaTransaccion')       .send({  CuentaOrigen:"37747947969500",  CuentaDestino:"055221948280621",       monto: 1000     })     .end( function(err,res){       if (err){         console.log(err);       } else {         console.log(res.body)       }  expect(res).to.have.status(404);     done();   } }); }); }); </pre>
<p>¿Qué pasaría si se desea obtener el reporte de débitos y créditos de la cuenta de un usuario en particular?</p> <p>¿Qué pasaría si el CUI enviado como parámetro de un usuario sí está relacionado con una cuenta en la base de datos? Según el método se esperaría un status 202.</p>	<pre> describe('Get transactions report: ',()=&gt;{   it('Should get user transactions report', (done) =&gt; {   chai.request(app)     .post('/reporteTransaccion')     .send({ cuenta: "37747947969500"})     .end( function(err,res){       if (err){         console.log(err);       } else {         console.log("Se ha obtenido el reporte de transacciones exitosamente.")       }     })   }) }); </pre>

	<pre> expect(res).to.have.status(202);     done();     }     });     });     }); </pre>
<p>¿Qué pasaría si se desea obtener el reporte de débitos y créditos de la cuenta de un usuario en particular?</p> <p>¿Qué pasaría si el CUI enviado como parámetro de un usuario no está relacionado con una cuenta en la base de datos? Según el método se esperaría un status 404.</p>	<pre> describe('Get transactions report: ',()=&gt;{     it('Shouldn't get user transactions report`, (done) =&gt; {     chai.request(app)     .post('/reporteTransaccion')     .send({ cuenta: "37747947969501"})     .end( function(err,res){         if (err){             console.log(err);         } else {             console.log("Error al obtener reporte de transacciones exitosamente.")         }          expect(res).to.have.status(404);         done();     }     });     });     }); </pre>
<p>Este método se encarga de generar la fecha actual y devolver en tipo String.</p>	<pre> describe("testeando generarFecha: ", function() {     describe("Validando fecha: ", function() {         it("Check the returned value using: assert.equal(value, 'value'): ", function() {             result = fecha.generarFecha();             assert.typeOf(result, 'string');         });     }); }); </pre>



## Frontend

Descripción	Prueba Unitaria
<p>Componentes utilizados en las paginas.</p> <p>Todos los componentes deben de crearse correctamente.</p>	<pre>describe('Test para los componentes de la aplicación.', ()=&gt;{   it('Menu se Renderizá sin error', ()=&gt;{     const wrapper = shallow(&lt;Menu/&gt;);     expect(wrapper).toHaveLength(1);   })    it('Menu Principal se Renderizá sin error', ()=&gt;{     const wrapper = shallow(&lt;MenuPrincipal/&gt;);     expect(wrapper).toHaveLength(1);   })    it('Formulario Login se Renderizar sin error', ()=&gt;{     const wrapper = shallow(&lt;FormLogin/&gt;);     expect(wrapper).toHaveLength(1);   })    it('Formulario Registro se Renderizá sin error', ()=&gt;{     const wrapper = shallow(&lt;FormularioRegistro/&gt;);     expect(wrapper).toHaveLength(1);   })    it('Pdf se Renderizá sin error', ()=&gt;{     const wrapper = shallow(&lt;PDF/&gt;);     expect(wrapper).toHaveLength(1);   }) })</pre>
<p>Formularios en la pagina de login y registro.</p> <p>Los formularios deben de aparecer en el centro de la pagina</p>	<pre>describe('Formularios creados correctamente.', ()=&gt;{   it('Formulario Login debe centrar su contenido.', ()=&gt;{     expect(shallow(&lt;FormLogin/&gt;).is('.text-center')).toBe(true);   })    it('Formulario Registro debe centrar su contenido.', ()=&gt;{     expect(shallow(&lt;FormularioRegistro/&gt;).is('.text-center')).toBe(true);   }) })</pre>
<p>El menu debera de contener el titulo correcto.</p>	<pre>describe('Debe aparecer el titulo de la pagina', ()=&gt;{   it('Debe tener el titulo analisis', () =&gt;{     expect(true);     render(&lt;Menu/&gt;);   }) })</pre>

```

expect(screen.queryByText(/analisis/i)).toBeInTheDocument();
})
it('Debe tener el titulo diseño', () =>{
  expect(true);
  render(<Menu/>);

  expect(screen.queryByText(/diseño/i)).toBeInTheDocument();
})
});

```

La pagina de perfil tiene que mostrar los datos correctos.

```

describe('Pagina profile', ()=>{
  it('debe mostrar texto: mis datos', () =>{
    render(<Profile/>)
    // Debe tener label de datos
    expect(screen.queryByText(/mis
datos/i)).toBeInTheDocument()
  })

  it('Debe tener boton transferencia', () =>{
    render(<Profile/>)

    expect(
      screen.getByRole(
        'button',
        {
          name: /realizar transferencia/i
        }
      )
    ).toBeInTheDocument();
  })

  it('Debe tener boton saldo', () =>{
    render(<Profile/>)

    expect(
      screen.getByRole(
        'button',
        {
          name: /saldo/i
        }
      )
    ).toBeInTheDocument();
  })

  it('Debe tener boton reporte', () =>{
    render(<Profile/>)

    expect(

```

```

        screen.getByRole(
            'button',
            {
                name: /generar reporte/i
            }
        )
    ).toBeInTheDocument();
})

it('Debe mostart nombre', () =>{
    let a = render(<Profile/>)
    // /expresion regular/ i -> ignore case

expect(screen.queryByText(/nombre/i)).toBeInTheDocument()
})

it('Debe mostart apellido', () =>{
    let a = render(<Profile/>)
    // /expresion regular/ i -> ignore case

expect(screen.queryByText(/apellido/i)).toBeInTheDocument()
})

it('Debe mostart cui', () =>{
    let a = render(<Profile/>)
    // /expresion regular/ i -> ignore case
    expect(screen.queryByText(/cui/i)).toBeInTheDocument()
})

it('Debe mostart correo', () =>{
    let a = render(<Profile/>)
    // /expresion regular/ i -> ignore case

expect(screen.queryByText(/correo/i)).toBeInTheDocument()
})
});

```

El registro debe de mostrar los labels indicados y el boton para registrar.

```

describe('Pagina registro', ()=>{
    it('debe de pedir nombre', () =>{
        render(<Registro/>)
        // Debe tener label de datos

expect(screen.queryByText(/nombre/i)).toBeInTheDocument()
})

    it('debe tener textbox de nombre', () =>{
        render(<Registro/>)

        expect(
            screen.getByRole(
                'textbox',

```

```

        { name: /nombre/i }
      )
    ).toBeInTheDocument();
  })

  it('debe de pedir apellido', () =>{
    render(<Registro/>)
    // Debe tener label de datos

    expect(screen.queryByText(/apellido/i)).toBeInTheDocument()
  })

  it('debe tener textbox de apellido', () =>{
    render(<Registro/>)

    expect(
      screen.getByRole(
        'textbox',
        { name: /nombre/i }
      )
    ).toBeInTheDocument();
  })

  it('debe de pedir cui', () =>{
    render(<Registro/>)
    // Debe tener label de datos
    expect(screen.queryByText(/cui/i)).toBeInTheDocument()
  })

  it('debe tener textbox de cui', () =>{
    render(<Registro/>)

    expect(
      screen.getByRole(
        'spinbutton',
        { name: /cui/i }
      )
    ).toBeInTheDocument();
  })

  it('debe de pedir saldo', () =>{
    render(<Registro/>)
    // Debe tener label de datos

    expect(screen.queryByText(/saldo/i)).toBeInTheDocument()
  })

  it('debe tener textbox de saldo', () =>{
    render(<Registro/>)

    expect(
      screen.getByRole(

```

```

        'spinbutton',
        { name: /saldo/i }
    )
    ).toBeInTheDocument();
})

it('debe de pedir correo', () =>{
    render(<Registro/>)
    // Debe tener label de datos

expect(screen.queryByText(/correo/i)).toBeInTheDocument()
})

it('debe tener textbox de correo', () =>{
    render(<Registro/>)

    expect(
        screen.getByRole(
            'textbox',
            { name: /correo/i }
        )
    ).toBeInTheDocument();
})

it('debe de pedir password', () =>{
    render(<Registro/>)
    // Debe tener label de datos

expect(screen.queryByText(/password/i)).toBeInTheDocument()
})

it('Debe tener boton de registro', () =>{
    render(<Registro/>)

    expect(
        screen.getByRole(
            'button',
            {
                name: /registrarse/i
            }
        )
    ).toBeInTheDocument();
})
});

```

Debe de aparecer el label "no. de cuenta".

```

describe('Pagina profile', ()=>{
    it('debe poder publicar', () =>{
        render(<Profile/>)
        // Debe tener label de datos
        expect(screen.queryByText(/no. de

```

```
cuenta:/i)).toBeInTheDocument()  
  })  
});
```

## Anexos

### Capturas de pantalla

#### *Pagina de login*

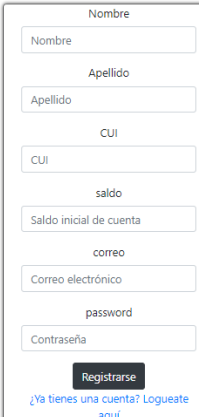
ANALISIS Y DISEÑO 1



A login form mockup featuring a circular profile picture placeholder with a blue silhouette. Below it, the label 'Usuario' is followed by a text input field containing 'No. de Cuenta'. Underneath, the label 'Contraseña' is followed by a text input field containing 'Contraseña'. At the bottom, there is a dark 'Iniciar Sesión' button and a blue link 'Regístrate ahora'.

#### *Pagina de registro*

ANALISIS Y DISEÑO 1



A registration form mockup with multiple text input fields. The fields are labeled: 'Nombre' (containing 'Nombre'), 'Apellido' (containing 'Apellido'), 'CUI' (containing 'CUI'), 'saldo' (containing 'Saldo inicial de cuenta'), 'correo' (containing 'Correo electrónico'), and 'password' (containing 'Contraseña'). At the bottom, there is a dark 'Registrarse' button and a blue link '¿Ya tienes una cuenta? Logueate aqui'.

#### *Pagina de perfil*

Oscar Eduardo Mazariegos López

No. de Cuenta: 91507698342290

Mis datos:

Nombre: Oscar Eduardo

Apellido: Mazariegos López

CUI: 297853160101

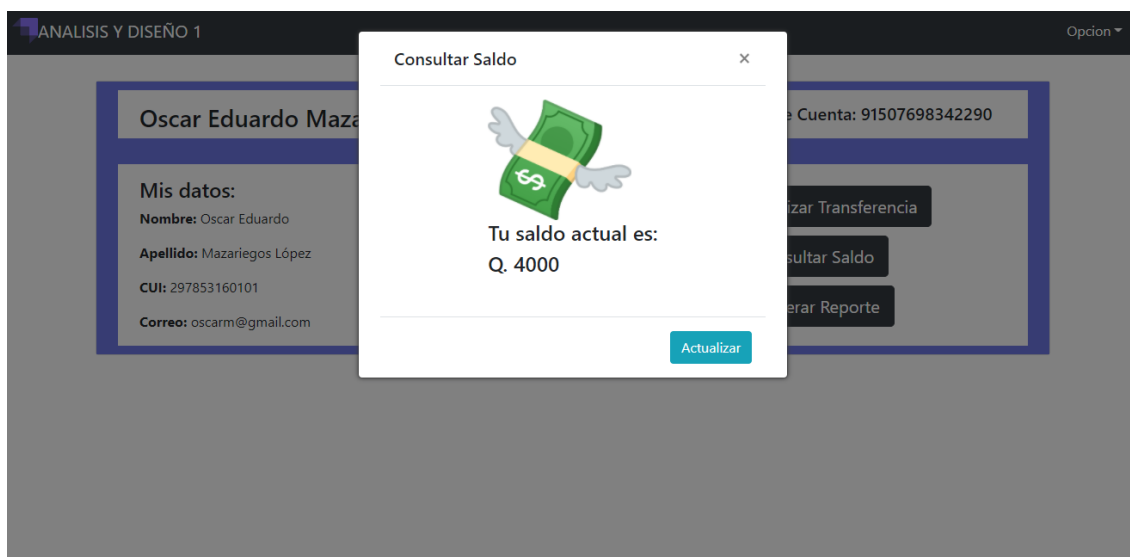
Correo: oscarm@gmail.com

Realizar Transferencia

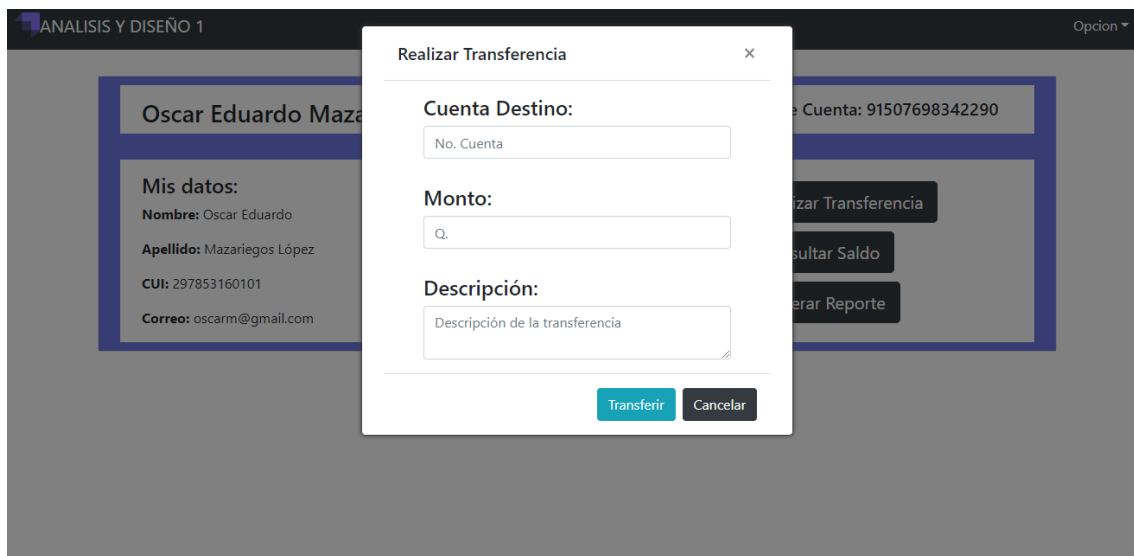
Consultar Saldo

Generar Reporte

*Ventana para consultar saldo.*



*Ventana para realizar transferencia.*



***Ventana para generar reporte.***

