



Universidad de Ingeniería y Tecnología

Ciencias de la Computación

Teoría de la Computación
Prof. José Fiestas

Proyecto

Ping-Pong

Autores

Lazo David
Barrios Alonso
Salazar Christian

Ping-Pong

Lazo David & Barrios Alonso & Salazar Christian

Índice

1. Introducción	2
2. Desarrollo	2
3. Automata	6
4. Código	7
5. Conclusión	7
6. Referencias	7

1. Introducción

El proceso de Login puede parecer algo sencillo a simple vista, solo escribes tu usuario y contraseña y listo, lograste ingresar. Pero en verdad, ¿sabes todo el proceso que el programa toma para que puedas disfrutar de los programas que usas, tales como Facebook, Instagram, entre otros?. Mediante este proyecto se simulará mediante autómatas el proceso que de "Login" "Sign Up".

Adicionalmente se simulará el proceso de comunicación entre computadoras llamado "Ping-Pong", el cual inicia con un estado de espera de mensaje, seguido de un estado de recibo de mensaje. Una vez realizado, se envía la confirmación del recibo. Luego se ejecuta el requerimiento de información, el cual, si le es denegado, se vuelve a enviar. Si es aceptado, regresa al estado inicial a esperar por un requerimiento de otra computadora.

2. Desarrollo

1. **User:** Para la clase usuario se declaró que sus atributos serían Nombre, Apellido, Username y Password, con el fin de crear un vector de usuarios, el cual funcionará como una base de datos.

```
struct User {  
    string Nombre;  
    string Apellido;  
    string Username;  
    string Password;  
};
```

2. **Log in:** Esta función se encarga de pedir el Username y el Password de un usuario previamente registrado, con la función CheckLogin() se encarga de verificar si en la base de datos existe un usuario el cual coincida con el username y el password ingresados. Si encuentra el usuario en la base de datos, este le enviará a la función PingPong(), caso contrario este te imprimirá un error y tendrá que volver a Loggearse.

```
bool Check_Log_in(string username, string password) {  
    auto comprobar = [username, password](User u) {  
        if(username == u.Username && password == u.Password)  
            return true;  
        return false;  
    };  
  
    auto it = find_if(users.begin(), users.end(), comprobar);  
  
    if(it != users.end())  
        return true;  
  
    else
```

```

        return true;
    }

    void Sign_in() {
        cout << "\n\n—————Sign_in—————\n";

        string temp_username, temp_password;

        cout << "Ingrese_usuario:_";
        cin >> temp_username; cin.ignore();

        cout << "\nIngrese_contrasena:_";
        cin >> temp_password; cin.ignore();

        if(Check_Log_in(temp_username, temp_password) == false)
            Ping_Pong(temp_username);

        else {
            cerr << "Usuario_y/o_contrasena_incorrecta\n";
            Sign_in();
        }
    }
}

```

3. **Sign up:** Esta función permite registrar a un usuario nuevo. Lo que el programa exige es que no puede haber dos usuarios con el mismo Username, por ende la función ComprobarUsuario() se encarga de comprobar que no exista otro username igual en la base de datos. Al verificar, este te enviará a la función SignIn() previamente vista, caso contrario te pedirá en username adecuado.

```

bool Comprobar_Usuario(string username) {
    auto comprobar = [username](User u) {
        if(username == u.Username)
            return true;
        return false;
    };

    auto it = find_if(users.begin(), users.end(), comprobar);

    if(it != users.end())
        return false;

    else
        return true;
}

void Sign_up() {
    cout << "\n\n—————Sign_up—————\n";
    User user;
    string temp_username;

    cout << "Ingrese_nombre:_";
    getline(cin, user.Nombre, '\n');
}

```

```

        cout << "\nIngrese_apellido:_";
        getline(cin, user.Apellido, '\n');

        do {
            cout << "\nIngrese_nombre_de_usuario:_";
            cin >> temp_username; cin.ignore();
        } while(Comprobar_Usuario(temp_username) == false);

        user.Username = temp_username;

        cout << "\nIngrese_su_contrasena:_";
        cin >> user.Password; cin.ignore();

        users.push_back(user);
        sort_users();

        Ping_Pong(user.Username);
    }

```

4. **Ping-Pong:** Esta función se encarga de realizar el juego de Ping-Pong con dos computadoras. Primero te da la bienvenida, esto con el fin de comprobar que se halla hecho un correcto Log in. Luego se le pide el número de saques, el cual será el número de iteraciones que se realizarán entre las computadoras. Por último te pide el valor inicial el cual jugará con las computadoras multiplicando el número dado por 2. Para finalizar, este te pide si deseas seguir continuando, si decides que sí, te mantendrás en tu cuenta jugando Ping-Pong, caso contrario, hará un Log out, el cual te devolverá a la pantalla de inicio.

```

void Ping_Pong(string username) {
    size_t saques;
    int number;

    cout << "\nBienvenido_" << username << "\n\n";

    cout << "Empezando_juego...\n\n";

    cout << "Ingrese_numero_saques:_";
    cin >> saques; cin.ignore();

    cout << "\nIngrese_numero_inicial:_";
    \includegraphics[width=0.4\linewidth]{Utec}

    for(; saques > 0; saques--) {
        cout << "Computador_1_==>_" << number << endl;

        number *= 2;
        cout << "\t\t\t\t\t" << number << " <==_Computador_2\n";
    }

    char option;
    cout << "Desea_volver_a_jugar_[s/n]:_";

```

```
cin >> option; cin.ignore();

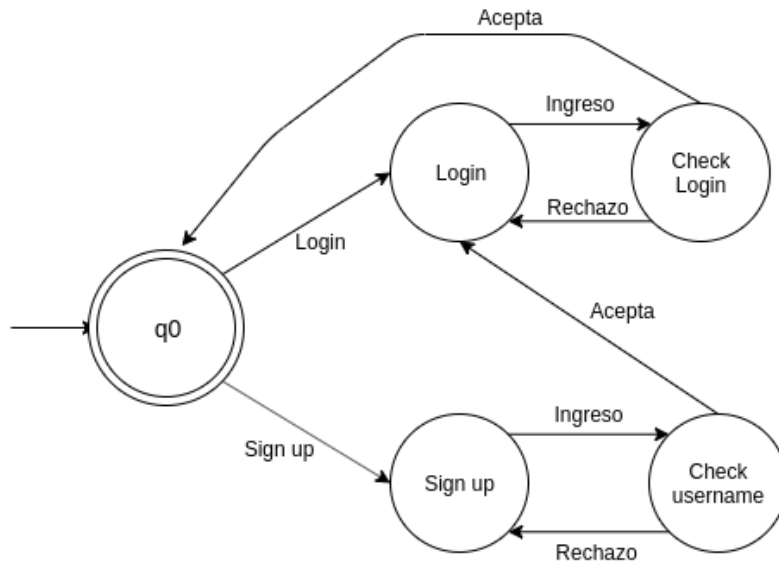
switch(option) {
    case 's':
        Ping_Pong(username);
        break;

    case 'n':
        Sign();
        break;

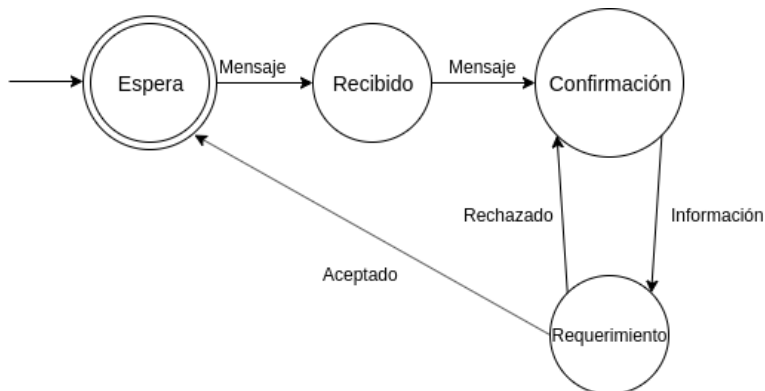
    default:
        cout << "No es una opcion valida\n";
}
}
```

3. Automata

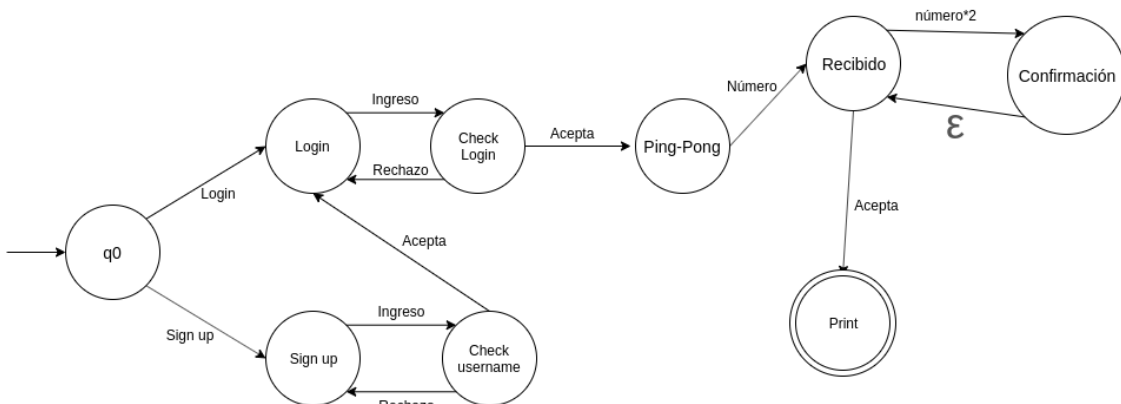
3.1. Login



3.2. Comunicacion entre dos computadoras



3.3. Ping-Pong



4. Código

Aquí se hizo el proceso de Ping-Pong

<https://github.com/alonso1311/TCproyecto.git>

5. Conclusión

En otras palabras básicamente que el inicial es el CIN, estado q1 y q2 son el sign in y sign up, el q3 y q4 eran las funciones una de ver si ya existe el username y la otra de ver si está correctamente escrito el usuario que ya existe, el R es el loop de rechazo hasta que se acepte y lleguen al estado q5 que es cuando ya se acepta y entra a la página web.

6. Referencias

- [1] C++ Standard Library Quick Reference. Peter Van Weert, Marc Gregoire. Apress, Jun 13, 2016.
- [2] Mastering the C++17 STL. Arthur O'Dwyer. Packt Publishing Ltd, Sep 28, 2017.
- [3] C++17 STL Cookbook. Jacek Galowicz. Jun 28, 2017 by Packt Publishing Ltd.
- [4] Competitive Programming. Steven Felix Halim, 2010
- [5] C++ Standard Library Quick Reference. Peter Van Weert, Marc Gregoire. Apress, Jun 13, 2016.
- [6] Introduction to the Theory of Computation, 3ra Edition, Michael Sipser, 2013.