

Descripción técnica del proyecto

Programa de operaciones de conjuntos

El programa se desarrolló en C++ y su interfaz es de texto.

Se utilizan dos librerías:

```
<iostream>  
<vector>
```

Se desarrollaron 12 funciones:

```
int bienvenida();  
void pideConjunto(vector<char> &universo, string tipo);  
void imprimeConjunto(vector<char> res, int t);  
void mensajeError(int index);  
int menuDeOpciones(vector<char> a, vector<char> b, vector<char> u, vector<char>  
&res);  
void revisaConjunto(vector<char> universo, vector<char> &conjunto, string tipo);  
void interseccion (vector<char> a, vector<char> b, vector<char> &res);  
void unionFn (vector<char> a, vector<char> b, vector<char> &res);  
void complemento (vector<char> vec, vector<char> universo, vector<char> &res);  
void diferencia (vector<char> vec1, vector<char> vec2, vector<char> &res);  
void potencia(vector<char> vec, vector<char> &res);  
vector<int> convertBinary(int num);
```

Descripción de funciones:

- *int bienvenida();*
Función que imprime menú principal, solicita input del usuario.
- *void pideConjunto(vector<char> &universo, string tipo);*
Función que solicita conjunto y se lo asigna a un vector por referencia.
- *void imprimeConjunto(vector<char> res, int t);*
Función que imprime conjunto, separado por espacios.
- *void mensajeError(int index);*
Función que imprime un mensaje de error, de acuerdo al índice argumento.

- *int menuDeOpciones(vector<char> a, vector<char> b, vector<char> u, vector<char> &res);*
Función que imprime posibles operaciones, recibe input de usuario y lleva a cabo operación seleccionada.
- *void revisaConjunto(vector<char> universo, vector<char> &conjunto, string tipo);*
Función para verificar que el conjunto argumento sea subconjunto del conjunto universo. En caso de no serlo, imprime error y vuelve a pedir conjunto.
- *void interseccion (vector<char> a, vector<char> b, vector<char> &res);*
Función que lleva a cabo operación intersección entre los conjuntos argumentos, modifica al vector resultado por referencia.
- *void unionFn (vector<char> a, vector<char> b, vector<char> &res);*
Función que lleva a cabo operación unión entre los conjuntos argumentos, modifica al vector resultado por referencia.
- *void complemento (vector<char> vec, vector<char> universo, vector<char> &res);*
Función que lleva a cabo operación complemento entre el conjunto argumento y el universo, modifica al vector resultado por referencia.
- *void diferencia (vector<char> vec1, vector<char> vec2, vector<char> &res);*
Función que lleva a cabo operación diferencia entre los conjuntos argumentos, modifica al vector resultado por referencia.
- *void potencia(vector<char> vec, vector<char> &res);*
Función que lleva a cabo operación potencia del conjunto argumento, modifica al vector resultado por referencia.
- *vector<int> convertBinary(int num);*
Función que convierte un número decimal a binario, utilizada para la operación potencia.