PROGRAMA PRINCIPAL

```
while (opcion != 'S') {
        cout << "Opciones eupt seleccion" << endl;
        cout << "1.- Inscribir ESTUDIANTE" << endl;
        cout << "2.- Borrar ESTUDIANTE" << endl;
        cout << "S.- Salir" << endl;
        cin >> opcion;
        switch (opcion){
                case '1':
                        inscripcion();
                        break;
                case '2':
…
```

TAD estudiantes

INTERFAZ

```
template<typename V> struct estudiantes;
template<typename V> void crear(estudiantes<V>& est);
template<typename V> int cardinal(const estudiantes<V>& est);
template<typename V> bool esVacia(const estudiantes<V>& est);
template<typename V> bool pertenece(const estudiantes<V>& est, const string c);
template<typename V> bool obtenerValor(const estudiantes<V>& est, const string c, V& v);
template<typename V> bool enMatricula(const estudiantes<V>& est);
template<typename V> bool anyadir(estudiantes<V>& est, const string c, const V& v);
template<typename V> bool quitar(estudiantes<V>& est, const string c);
template<typename V> void cerrarInscripcion(estudiantes<V>& est);
template<typename V> int pasarTurno(estudiantes<V>& est);
template<typename V> int obtenerCandidatoSuClave(const estudiantes<V>& est, string& c);
template<typename V> int obtenerCandidatoSuValor(const estudiantes<V>& est, V& v);
template<typename V> int actualizarCandidato(estudiantes<V>& est, const V& v);
template<typename V> int eliminarCandidato(estudiantes<V>& est);
template<typename V> void listar(estudiantes<V>& est);
template<typename V> void iniciarIterador(estudiantes<V>& est);
template<typename V> bool existeSiguiente(const estudiantes<V>& est);
template<typename V> bool siguiente(estudiantes<V>& est, string& c, V& v);
```

VERIFICAR OPERACIONES

operaciones *{se requiere que estén definidas las siguientes operaciones de comparación y de transformación a cadena:}*

generaCadena: valor v → cadena

OPCIONES PARA "generaCadena"

Utilizar ostringstream de librería <sstream>

```
#include <sstream>

ostringstream cadena_listado;

cadena_listado << "dni: " << dni;

cadena_listado << "nombre: " << nombre(p);

return cadena_listado.str();
```