

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE
PRUEBAS DE SOFTWARE

NOMBRE: Alisson Clavijo

NRC: 9870

A continuación, se muestra un procedimiento en pseudocódigo:

```
static inline bool es_cedula_valida(std::string cedula) {
    std::regex regex_cedula("^\\d{10}$");

    if (!std::regex_match(cedula, regex_cedula)) {
        return false;
    } else if (cedula.length() != 10) {
        return false;
    }

    int digito_verificador = std::stoi(cedula.substr(9, 1));
    int suma = 0;
    int digito = 0;

    for (int i = 0; i < 9; i++) {
        digito = std::stoi(cedula.substr(i, 1));

        if (i % 2 == 0) {
            digito *= 2;

            if (digito > 9) {
                digito -= 9;
            }
        }

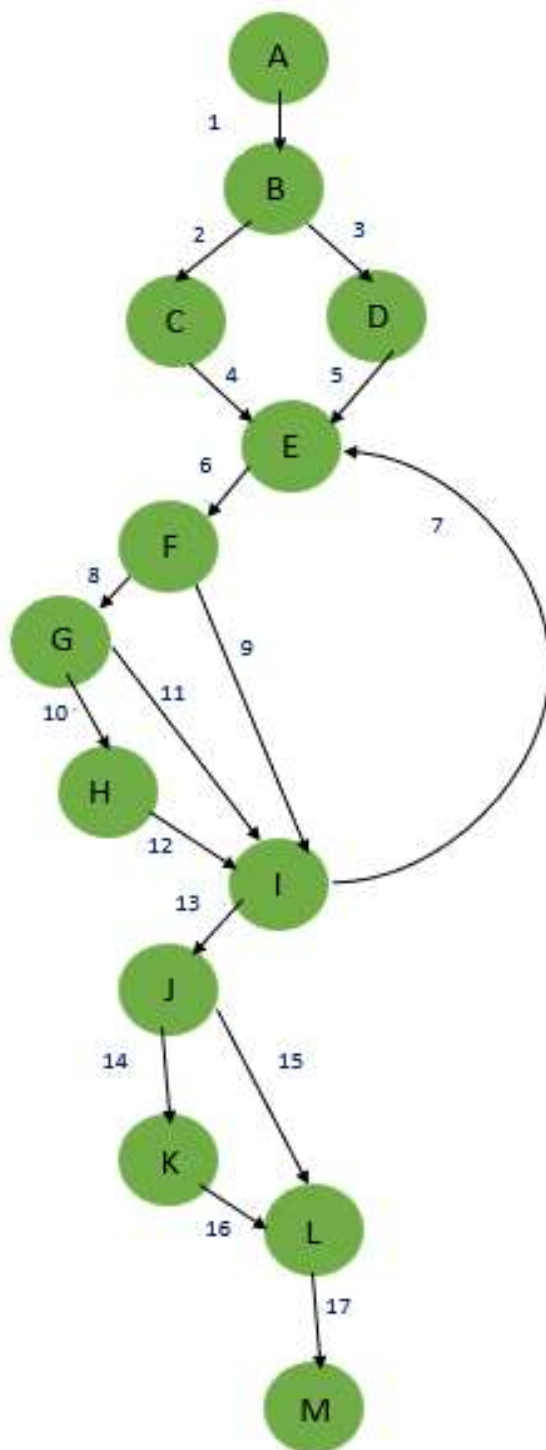
        suma += digito;
    }

    int superior = (10 - (suma % 10)) + suma;

    if (suma % 10 == 0) {
        return digito_verificador == 0;
    }

    return (superior - suma) == digito_verificador;
}
```

Construir el Grafo (nodos y aristas) respecto al código anterior



1) Obtener todos los caminos posibles que usted encuentre

- a. Camino 1: 1,2,4,6,8,10,12,13,14,16,17
- b. Camino 2: 1,2,4,6,8,10,12,13,15,17
- c. Camino 3: 1,2,4,6,8,11,13,14,16,17
- d. Camino 4: 1,2,4,6,9, 7n
- e. Camino 5: 1,2,4,6,8,11,13,15,17
- f. Camino 6: 1,2,4,6,9,13,14,16,17
- g. Camino 7: 1,2,4,6,9,13,15,17
- h. Camino 8: 1,2,4,6,8,10,12,7,6,8,11,7,6,9,13,14,16,17
- i. Camino 9: 1,2,4,6,8,10,12,7,6,8,11,7,6,9,13,15,17
- j. Camino 10: 1,2,4,6,8,11,7,6,8,10,12,13,15,17
- k. Camino 11: 1,2,4,6,8,11,7,6,8,10,12,13,14,16,17
- l. Camino 12: 1,3,5,6,9,7,6,8,11,13,15,17
- m. Camino 13: 1,3,5,6,9,7,6,8,11,13,14,16,17
- n. Camino 14: 1,3,5,6,8,10,12,7,6,9,13,14,16,17
- o. Camino 15: 1,3,5,6,8,10,12,7,6,9,13,15,17

2) Calcular la complejidad ciclomática basada en caminos.

Complejidad: $17 - 13 + 2 = 6$

3) Construya todos los posibles casos de prueba que se pueden generar para probar los caminos encontrados.

Camino	Representa	Decisión
1,2,4,6,8,10,12,13,14,16,17	cédula válida con dígito verificador 0	Probar
1,3,5,6,8,10,12,7,6,9,13,15,17	cédula válida con dígito verificador no 0	Probar
1,2,4,6,8,11,7,6,8,10,12,13,15,17	cédula inválida (longitud incorrecta)	Probar
1,2,4,6,8,10,12,7,6,8,11,7,6,9,13,14,16,17	cédula inválida (suma no múltiplo de 10, dígito verificador incorrecto)	Probar
1,3,5,6,9,7,6,8,11,13,15,17	cédula inválida (patrón incorrecto)	Probar
1,2,4,6,8,10,12,13,14,16,17	cédula inválida (patrón correcto, pero dígito verificador incorrecto)	Probar
1,2,4,6,8,10,12,7,6,8,11,7,6,9,13,15,17	cédula válida con dígito verificador 0	Probar
1,2,4,6,8,11,7,6,8,10,12,13,14,16,17	cédula válida con dígito verificador no 0	Probar
1,2,4,6,8,11,7,6,8,10,12,13,14,16,17	cédula inválida (patrón correcto, pero dígito verificador incorrecto)	Probar

Camino No.	Caso de prueba	Objetivo	Resultado Esperado
1,2,4,6,8,10,12,13,14,16,17	cedula_valida_0 = "1234567890"	cédula válida con dígito verificador 0	cedula_valida_0 = "1234567890"
1,3,5,6,8,10,12,7,6,9,13,15,17	cedula_valida_no_0 = "9876543212"	cédula válida con dígito verificador no 0	cedula_valida_no_0 = "9876543212"
1,2,4,6,8,11,7,6,8,10,12,13,15,17	cedula_longitud_invalida = "123456789"	cédula inválida (longitud incorrecta)	cedula_longitud_invalida = "123456789"
1,2,4,6,8,10,12,7,6,8,11,7,6,9,13,14,16,17	cedula_caracteres_invalidos = "A234567890"	cédula inválida (suma no múltiplo de 10, dígito verificador incorrecto)	cedula_caracteres_invalidos = "A234567890"
1,3,5,6,9,7,6,8,11,13,15,17	cedula_patron_invalido = "44444444A1"	cédula inválida (patrón incorrecto)	cedula_patron_invalido = "44444444A1"
1,2,4,6,8,10,12,13,14,16,17	cedula_patron_correcto_digito_incorrecto = "5555555556"	cédula inválida (patrón correcto, pero dígito verificador incorrecto)	cedula_patron_correcto_digito_incorrecto = "5555555556"
1,2,4,6,8,10,12,7,6,8,11,7,6,9,13,15,17	cedula_valida_0 = "1234567890"	cédula válida con dígito verificador 0	cedula_valida_0 = "1234567890"
1,2,4,6,8,11,7,6,8,10,12,13,14,16,17	cedula_valida_no_0 = "9876543212"	cédula válida con dígito verificador no 0	cedula_valida_no_0 = "9876543212"
1,2,4,6,8,11,7,6,8,10,12,13,14,16,17	cedula_patron_correcto_digito_incorrecto = "5555555556"	cédula inválida (patrón correcto, pero dígito verificador incorrecto)	cedula_patron_correcto_digito_incorrecto = "5555555556"