

1. Introduccion

Este es un sistema simplificado que permite definir carreras de grado, junto con sus respectivas materias y planteles docentes, sobre las cuales estudiantes se puedan inscribir a través de sus libretas universitarias (las cuales tienen largo fijo). El sistema permite el agregado de docentes a los planteles de las materias, así como establecer un cupo sobre cada materia en base a la cantidad de docentes disponibles. Se estimó que debe haber a lo sumo 250 estudiantes por cada profesor, 100 estudiante por cada JTP, 20 estudiantes por cada Ay1 y 30 por cada Ay2.1 En caso de que haya más que esa cantidad de estudiantes por cada cargo docente, se considerará que el cupo está excedido. Por otro lado, como es sabido, una misma materia puede tener varios nombres, los cuales dependen de la carrera de grado. Por ejemplo: “Algoritmos y Estructuras de Datos 2” (para Lic. en Cs. de Datos) y “Algoritmos y Estructuras de Datos” (para Lic. en Cs. de la Computación)

Especificación del Sistema SIU Guaraní

1. **nuevoSistema(in infoMaterias: seq<InfoMateria>, in libretas-Universitarias: seq<string>): SistemaSIU**
Inicializa el sistema SIU Guaraní.

$$\text{Complejidad: } O\left(\sum_{c \in C} |c| \cdot |Mc| + \sum_{m \in M} \sum_{n \in N_m} |n| + E\right)$$

Donde:

- **ParCarreraMateria** es `<carrera: string, nombreMateria: string>`.
- **InfoMateria** es `seq<ParCarreraMateria>`.

2. **inscribir(inout sistema: SistemaSIU, in estudiante: string, in carrera: string, in nombreMateria: string)**
Inscribe al estudiante en la materia m de la carrera c .

$$\text{Complejidad: } O(|c| + |m|)$$

3. **inscriptos(in sistema: SistemaSIU, in carrera: string, in nombreMateria: string): N**
Dada una carrera c y una materia m , devuelve la cantidad de estudiantes inscriptos en la materia, incluyendo a los estudiantes de todas las carreras.

$$\text{Complejidad: } O(|c| + |m|)$$

4. **agregarDocente(inout sistema: SistemaSIU, in cargo: CargoDocente, in carrera: string, in nombreMateria: string)**
Agrega a un docente al plantel de la materia m de la carrera c , y al plantel de todas las otras materias que son sus equivalentes en el resto de las

carreras.

Complejidad: $O(|c| + |m|)$

5. plantelDocente(in sistema: SistemaSIU, in carrera: string, in nombreMateria: string): seq<N>

Devuelve un arreglo de enteros donde sus posiciones representan (en orden) la cantidad de profesores, JTPs, AY1 y AY2, respectivamente, para la materia m dada la carrera c .

Complejidad: $O(|c| + |m|)$

6. excedeCupo?(in sistema: SistemaSIU, in carrera: string, in nombreMateria: string): bool

Devuelve verdadero si la cantidad de inscriptos en la materia m de la carrera c excede el cupo.

Complejidad: $O(|c| + |m|)$

7. carreras(in sistema: SistemaSIU): seq<string>

Devuelve el listado de carreras de grado, ordenado lexicográficamente.

Complejidad: $O\left(\sum_{c \in C} |c|\right)$

8. materias(in sistema: SistemaSIU, in carrera: string): seq<string>

Devuelve el listado de materias de una carrera c , ordenado lexicográficamente.

Complejidad: $O\left(|c| + \sum_{mc \in Mc} |mc|\right)$

9. materiasInscriptas(in sistema: SistemaSIU, in estudiante: string): N

Devuelve la cantidad de materias en las que se encuentra inscripto/a un/a estudiante.

Complejidad: $O(1)$

10. cerrarMateria(inout sistema: SistemaSIU, in carrera: string, in nombreMateria: string)

A partir de una carrera c , cierra la materia m para todas las carreras de grado.

Complejidad: $O(|c| + |m| + |n| + E_m)$