

Correcciones Grupo *tortugasNinja*

Nota: **Aprobado**

Generales

- Pasa Los Tests: Muy Bien.
- Estructura: Muy Bien
- Calidad de Código: Muy Bien
- Invariantes: Bien

SistemaSIU.java

Tienen el siguiente fragmente repetido muchas veces:

```
Materia m = (Materia) ((Carrera) carreras.buscar(carrera)).getMaterias().buscar(materia);
```

Hubiera estado bueno que se hubieran hecho un método auxiliar que hiciera eso.

InvRep

- $|\text{SIU.alumnos.getclaves()}| = |\text{LibretasUniversitarias}|$

`LibretasUniversitarias` no está definido. ¿qué están queriendo comparar?

- No hay alumnos repetidos en `SIU.alumnos`.
- No hay carreras repetidas.

Estas dos son triviales, dado que los diccionarios nunca tienen claves repetidas.

- El array “`SIU.carreras.getClaves()`” esta ordenado lexicograficamente.
- El array “`SIU.carreras.buscar(carrera).getMaterias().getClaves()`” esta ordenado lexicograficamente.

Estas condiciones se cumplen independientemente de la instancia de `carreras`.

Lo que era importante poner era

para todas las claves de **alumnos**, su significado es igual a la cantidad de materias que se encuentran en **carreras** para las cuáles el alumno está en sus inscriptos.

Trie.java

InvRep

```
// Invariante de Representacion del Trie
/*
 * - Cada nodo, excepto la raiz, representa un caracter.
 */
// Invariante de Representacion del TrieNode
/*
 * - Cada nodo tiene un "valor" para indicar la terminacion de una palabra.
 */
```

Estos son los puntos que están mal. Están mal porque son consideraciones semánticas y no están hablando de cuándo una instancia es válida o no.

No hay dos hijos de un mismo nodo con el mismo caracter.

Este punto está mal porque es algo que está implícito en la estructura y no requiere aclaraciones adicionales, es decir, en una posición de un arreglo solo puedo guardar un único valor.

Falta agregar “no hay nodos inútiles”.

obtenerClaves()

Ojo con obtener claves, está mal la complejidad.

Cuando realizan **prefijo + (char) i**, esta operación es $O(|\text{prefijo}|)$. Como lo hacen en cada paso, finalmente les queda una complejidad cuadrática con respecto al largo de la clave.

Deberían usar una estructura de pila para ir apilando los caracteres y desapilando al salir. Luego construyen el string únicamente cuando llegan a una definición.