

# Correccion - Gonzalez Calderon, Julian - TP Hash:

Segundo cuatrimestre 2021

---

Esta es la correccion de la primer entrega del TP de Hash, en ella vas a encontrar comentarios generales, sobre tu codigo, sobre el informe que entregaste y finalmente sugerencias para futuras entregas. Al final del documento, vas a encontrar tu nota. Ya casi estamos en el final de la materia, por lo tanto, recuerda que algunas correcciones que se arrastran de antes y no eran tan graves, empiezan a ser mas notorias; acordate que la idea es ir mejorando en base a las correcciones hechas anteriormente. Si surgen cosas nuevas, es normal debido al aumento de complejidad en los temas, asi que no te preocupes si ves mas comentarios. Como siempre, te pedimos que le prestes atencion a la devolucion del TP. Me interesa que leas los comentarios independientemente de la nota final, que al final del dia, es solo un numero :)

## En general:

- Se cumple la mayoria de buenas practicas de codigo.
- En general el codigo esta prolijo y limpio, buen trabajo!
- Podrian sumarse algunos comentarios en algunas secciones del codigo.
- Muy buenas pruebas.

## README:

- Recuerda que el Hash no guarda solo valores, sino PARES de [clave, valor].
- Hash abierto: Ojo aca con la redaccion, que al principio parece que decis que el hash es abierto por que el elemento se encuentra en el indice de su clave. En todo caso por eso seria direccionamiento cerrado. Mas adelante se entiende mas, pero cuidado con como lo redactas en el informe.
- Aclaracion: Acordate que se pueden usar otros tds como estructuras externas en el abierto, no solo listas.
- Cuando creas varios archivos adicionales, para estructuras internas, como la lista, o hash\_interno, ademas de explicar la funcion, hace un breve comentario de que tiene cada archivo, o porque lo separaste asi.

## Código:

- Numeros Magicos y Codigos de retorno: Ojo con devolver un 0, recuerda que si te pasan un string null por ejemplo, y no podes hacer funcion de hash, lo que devolves deberia ser un error, que puede ser una constante, que a su vez, vale 0.
- Hay elementos que, mas que pertenecer a hash\_interno, deberian pertenecer a lista me parece., por ejemplo, insertar par en lista. Acordate que la responsabilidad de la funcion y las cosas con las que opera, te dicen en donde deberian ir.
- Capaz no valia la pena extraer cosas a lista\_interna, y podrias haber simplemente modificado lista en este caso, pero entiendo la idea de hacerlo asi por un posible futuro cambio y reutilizacion de la lista original.

## Pruebas:

- Parecido a lo de la otra vez, cuidado con los mensajes de las pruebas, por ejemplo decir "puedo modificar un elemento" y solo chequeaste si el hash lo contiene, ahí lo correcto sería algo tipo "El elemento que modifique no se perdió" pues devuelve true en contiene, pero hasta ahí no sabes si el cambio se realizó.
- Hubiera agregado algunas pruebas de estrés capaz, pero fuera de eso buen nivel de pruebas :)

### **Sugerencias:**

- Redacción, sobre todo en lo teórico. Ojo con como expresas las ideas en papel.
- Acordate de buscar el punto para la extracción a archivos, hay un momento donde capaz es exceso.

**Nota: 1/2. Buen TP! :)**