



PD7

ejercicio 1

hashCode Object: se devuelve la dirección interna del objeto pasada a Integer

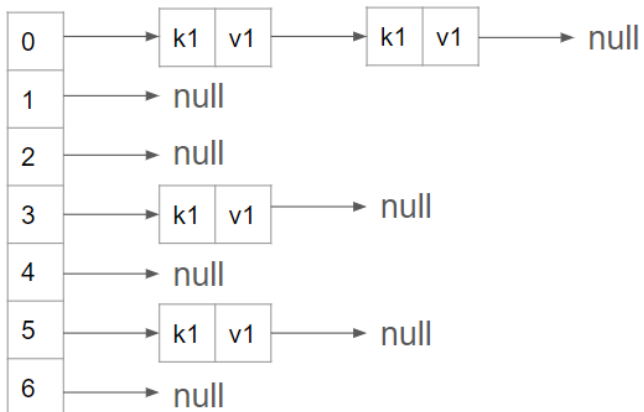
hashCode Integer: devuelve el valor primitivo *int* representado por este objeto Integer

hashCode String: devuelve $s[0]*31^{(n-1)} + s[1]*31^{(n-2)} + \dots + s[n-1]$

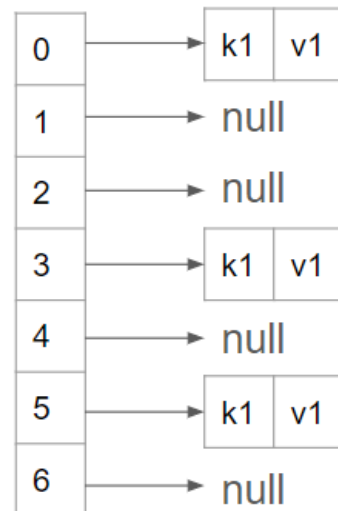
las implementaciones son distintas porque los enteros ya se distribuyen de manera uniforme, pero los strings hay que distribuirlos para evitar colisiones y esta fórmula lo permite.

ejercicio 2

con lista enlazada



sin lista enlazada



0	
1	
2	
3	
4	
5	HolaMundo
6	

7	HashMap
8	Hola
9	Colecciones

ejercicio 3

```
public class Alumno : Object {
    private int id;
    private String fullName;
    private String email;

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null) return false;
        if (this.getClass() != o.getClass()) return false;
        User user = (User) o;
        return id == user.id
            && (fullName.equals(user.fullName)
            && email.equals(user.email));
    }

    @Override
    public int hashCode() {
        int hash = 7;
        hash = 31 * hash + (int) id;
        hash = 31 * hash + (name == null ? 0 : name.hashCode());
        hash = 31 * hash + (email == null ? 0 : email.hashCode());
        return hash;
    }
}
```

el método hashCode combina los valores hash de los atributos del objeto para que se refleje todo el contenido del mismo