

Universidad de La Frontera

Facultad de Ingeniería y Ciencias

Departamento de Ciencias de la Computación e Informática

Ingeniería Informática

**Prueba 1: Generala**

2020-08-07

Integrantes : Héctor Sepúlveda G.

Asignatura : Programación Avanzada

Profesor : Marcelo Esperguel.

[1° commit: “Proyecto original: primer commit” 3](#__RefHeading___Toc383_4175283450)

[2° commit: “Metodos largeStraight, thereIsRepetitiveDice y thereIsADice” 3](#__RefHeading___Toc854_2584329060)

[3° commit: “Modificación de método Generala:chance" 4](#__RefHeading___Toc492_2911584482)

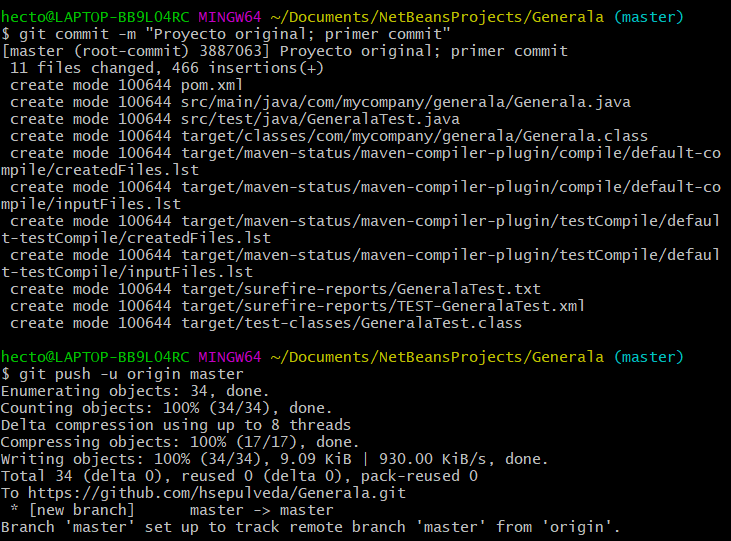
[4° commit: ”Modificación de metodo Generala:generala ” 4](#__RefHeading___Toc494_2911584482)

[5° commit: ”Metodo addACertainNumber para ones, twos, threes, fours, fives y sixes” 4](#__RefHeading___Toc496_2911584482)

[6° commit: ”Modificacion smallStraight” 5](#__RefHeading___Toc498_2911584482)

# 1° commit: “Proyecto original: primer commit”

Se descargó el código a mejorar desde campus virtual y se hizo el primer commit con el proyecto original.



En esta etapa no se efectuó ningún cambio.

# 2° commit: “Metodos largeStraight, thereIsRepetitiveDice y thereIsADice”

En esta etapa se completó el metodo largeStraight que debe cerciorarse de entregar el puntaje correcto, para lo cual se deben cumplir 2 condiciones:

1) Que los dados no sean repetidos.

2) Que no haya un 1.

Debido a estas 2 condiciones nacen 2 métodos que se preocupan de ver si se cumplen o no:

1. thereIsRepetitiveDice, el cual recibe 5 dados y revisa la primera condición.
2. ThereIsADice, el cual recibe 5 dados y un numero “target” para ver si se encuentra entre los 5 dados. Este método es bastante general, por lo que es candidato a ser reutilizado donde se necesite.

El método largeStraight usa ambos métodos anteriores y si se da alguna de las 2, devuelve 0; De lo contrario devuelve 20.

Implementar todo en el metodo largeStraight hubiera generado smells, por lo que inmediatamente se optó por separar las tareas en métodos adicionales.

# 3° commit: “Modificación de método Generala:chance"

Se ha modificado el metodo chance de la clase Generala el cual usaba la expresión “total+=” en exceso, usando esta variable como contador fuera de un ciclo, lo cual se sustituyó con programación funcional, acotando el código y solucionando el smell.

# 4° commit: ”Modificación de metodo Generala:generala ”

Se ha modificado el método “generala” para que trabaje con programación funcional, acotando el código y haciéndolo más entendible, gracias a la incorporación del método Generala:areAllTheSame, el cual es un método boolean que recive un VarArg y determina sí todos los datos tienen el mismo valor.

# 5° commit: ”Metodo addACertainNumber para ones, twos, threes, fours, fives y sixes”

Se han modificado los métodos ones, twos, threes, fours, fives y six, los cuales tenían un código difícil de comprender, con la inclusión de 2 métodos polimórficos llamados addACertainNumber, los cuales hacen lo mismo para todos los métodos nombrados: Sumar un determinado número. Ante la inhabilidad de modificar la entrada de los métodos para no alterar la prueba, se han creado 2 métodos que hacen exactamente lo mismo, pero dependiendo de los parámetros, una será utilizada por ones, twos y threes y el otro método por fours, fives y sixes.

# 6° commit: ”Modificacion smallStraight”

Siguiendo con la estrategía usada con el metodo largeStraight, se eliminó la programación repetitiva e innecesaria, se utilizó programación funcional y se utilizaron los 2 métodos hechos para el metodo largeStraight.

Link Repositorio: <https://github.com/hsepulveda/Generala>