



IIC3745 - Testing (2020 / II)

Actividad 5 - Solución

El objetivo de la actividad era que los/as alumnos/as se enfrentarían a una batería de tests promedio y la mejoraran mediante el uso de *mutant*. La corrección consiste de dos ítems.

Coverage

[Este link](#) contiene el repositorio de solución (equivalente al repositorio base + dos *commits*) para correr el script / tests / *mutant* y obtener el coverage de mutación. El/la estudiante debió haber utilizado este como base para hacer las nuevas pruebas y debió entregar sus archivos de la siguiente manera:

```
.
├── app/
├── spec/
│   └── dcc_burger_spec.rb
└── integrantes.md
```

La evaluación del coverage se hace copiando estos archivos en el repositorio para luego correr *mutant* dentro del contenedor de Docker. **Si algún archivo no se encuentra en la ubicación pedida, aplicar descuento de 5 décimas**, pero moverlos y corregir de todos modos.

Código

Dados los cambios que hace *mutant*, a veces las pruebas adicionales no son suficientes para asegurarnos de tener 100% de coverage de mutación. Por esto, se debía recurrir a un cambio de código sin alterar la lógica de este. En particular, **estaba prohibido modificar**:

- Mensajes de excepciones o las condiciones para levantarlas.
- Precios de ingredientes, hamburguesas o tipos de pan.
- Restricciones para cada tipo de hamburguesa (*CustomBurger* u *OriginalBurger*).
- Método de valorización o excepciones de la validación de dinero disponible.

Cualquier cambio de este tipo sin una buena justificación resulta en un descuento. En el repositorio de la solución, existe un [commit](#) que muestra un posible conjunto de cambios válidos, desglosando:

1. *mutant* introduce la mutación a `.eq?` o `.equal?` por un detalle técnico de *Ruby*. El cambio de `==` a otro operador no omite casos bordes, por lo que su reemplazo a `.eq?`, `.equal?` o `===` es correcto.

10	-	<code>if type == 'custom'</code>
10	+	<code>if type === 'custom'</code>

Algo similar ocurre en la línea 79.

2. Dado que antes del `if / elsif` se hace una validación del tipo de hamburguesa, el `elsif` se hace irrelevante (i.e. nunca se llegará a ese código si la hamburguesa es de algún tipo distinto a `'custom'` u `'original'`). Por esto, el cambio de `elsif` a `else` es correcto

16	-	<code>elsif type == 'original'</code>
16	+	<code>else</code>

3. `StandardError` es el error que por defecto se levanta en *Ruby*, por lo que la declaración del tipo de excepción está demás en el código y *mutant* lo exige como cambio.

40	-	<code>raise StandardError, message</code>
40	+	<code>raise message</code>

Esto es un posible conjunto de cambios, lo importante es entender que no se debe cambiar la lógica de negocios del código. Tomar en cuenta el efecto de acciones previas, como en el caso de los tipos de hamburguesa y `validate_type`!