



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IIC2113 - Diseño Detallado de Software (2020-2)

## Ejercicio I2

### Indicaciones

- Pueden responder este ejercicio en C# o en Ruby. Recuerden que a lo menos uno de los dos ejercicios prácticos del semestre debe estar en C# (pueden responder una interrogación en Ruby y la otra en C#, o ambas en C#).
  - Deben subir el o los archivos comprimidos en un .zip
  - Todas las preguntas deben ser vía Issues de Github con el tag [Ejercicio I2 Práctico].
  - **El objetivo de este ejercicio es que tu solución sea modular, use buenas prácticas y sea entendible, además de funcional. El ejercicio consta de un total de 100 puntos, los cuales se reparten de la siguiente forma**
    - **Resuelve el problema correctamente [30 puntos]**
    - **El diseño de la aplicación obedece a los principios SOLID [40 puntos]**
    - **El diseño de la aplicación obedece a los principios fundamentales [10 puntos]**
    - **Uso de buenas prácticas [20 puntos]**
- En caso de no resolver el problema (no compila o se cae en ejecución) la nota máxima a la que puede optar es 5.0.**
- **Debe subir un readme con instrucciones de ejecución: comandos, herramientas usadas (vs code, visual studio code, etc), sistema operativo y breve explicación de implementación.**

Plazo de entrega: 27 de Noviembre a las 23:59 vía buzón de tareas en Canvas.

Crea una aplicación de consola que espere un input numérico del tipo entero que vaya desde el 1 al 10. Tu programa deberá generar tantos colores en hex como el input del usuario. El algoritmo para generar los hex queda a tu criterio, pero considera que deben ser colores válidos. Luego, utilizando la api de [The Color Api](#) deberás generar una paleta de 3 colores de tipo triada y mostrar en consola las 3 paletas mostrando el hex, rgb, hsl y cymk de cada color en las paletas. Luego de eso termina la ejecución del programa.

El formato para mostrar en la consola es el siguiente:

> 3

#1 schema:

```
hex: #735C2C,  
rgb: rgb(114, 47, 144)  
hsl: hsl(281, 51%, 37%)  
cmyk: cmyk(21, 67, 0, 44)
```

```
hex: #2E866D,  
rgb: rgb(46, 134, 109)  
hsl: hsl(163, 49%, 35%)  
cmyk: cmyk(66, 0, 19, 47)
```

```
hex: #79309A,  
rgb: rgb(121, 48, 154)  
hsl: hsl(281, 52%, 40%)  
cmyk: cmyk(21, 69, 0, 40)
```

#2 schema:

```
hex: #5748E1,  
rgb: rgb(87, 72, 225)  
hsl: hsl(246, 72%, 58%)  
cmyk: cmyk(61, 68, 0, 12)
```

```
hex: #E86B56,
```

rgb: rgb(232, 107, 86)  
hsl: hsl(9, 76%, 62%)  
cmyk: cmyk(0, 54, 63, 9)

hex: #66EE76,  
rgb: rgb(102, 238, 118)  
hsl: hsl(127, 80%, 67%)  
cmyk: cmyk(57, 0, 50, 7)

### #3 schema:

hex: #31DB39,  
rgb: rgb(77, 62, 227)  
hsl: (123, 70%, 53%)  
cmyk: cmyk(78, 0, 74, 14)

hex: #4D3EE3,  
rgb: rgb(231, 80, 69)  
hsl: hsl(245, 75%, 57%)  
cmyk: cmyk(66, 73, 0, 11)

hex: #EA584D,  
rgb: rgb(234, 88, 77)  
hsl: hsl(4, 79%, 61%)  
cmyk: cmyk(0, 62, 67, 8)