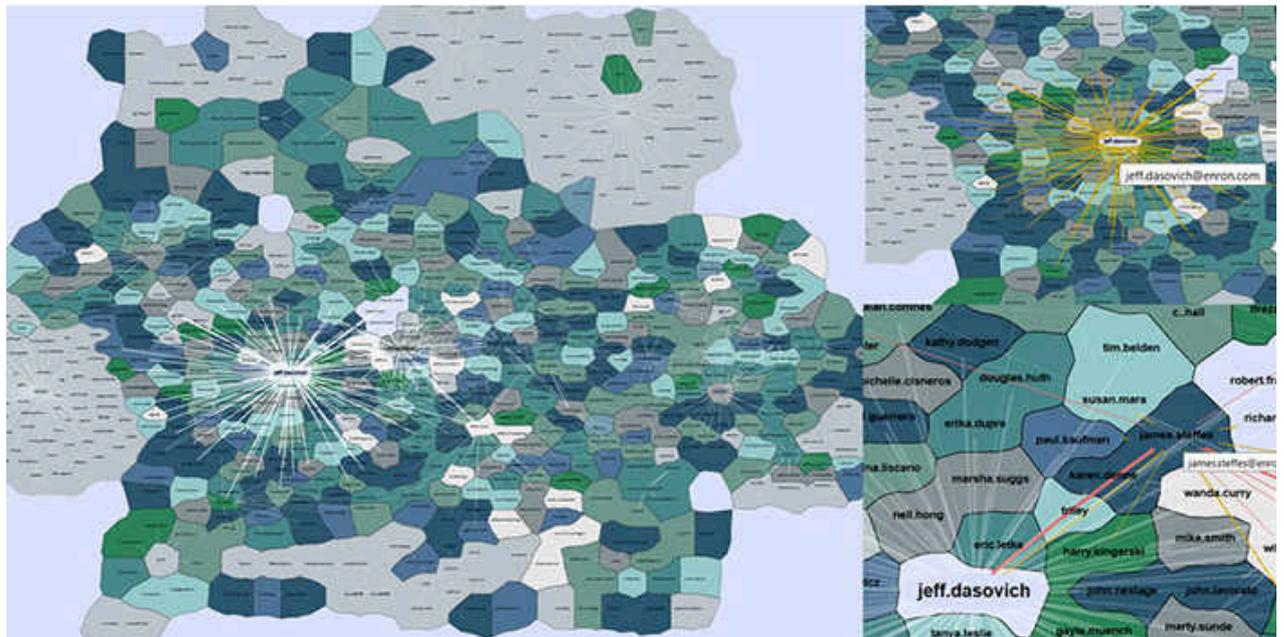


cooking skills: combinando colores con código

ph.d. franklin hernández-castro.

Instituto Tecnológico de Costa Rica
franhernandez@itcr.ac.cr



2da edición: enero 2022

como citar este trabajo:
Hernández-Castro, F. (2022). **cooking skills:
combinando colores con código**. Escuela de Diseño Industrial,
Instituto Tecnológico de Costa Rica. Cartago, Costa Rica.

resumen

esta nota técnica responde a las necesidades de mis estudiantes de combinar colores desde código.

la mayoría de los casos para controlar una composición cromática se usan pocos matices (a menudo solo uno), sin embargo hay veces en que es **necesario combinar un número indeterminado de matices** y cuando los datos son muchos es necesario que se haga por código. dicho de otro modo, es la computadora la que debe escoger los colores.

en este documento muestro el **modo más fácil y seguro de manejar una composición, inicialmente indeterminada, desde código.**

acerca de este tema desarrollé en el pasado algunas teorías más elaboradas que se pueden profundizar en mi tercer libro "el error de le corbusier" [1], pero en este documento me voy a referir a recetas rápidas de cómo combinar colores en "real-time".

índice

combinando colores	3
1. el problema	3
2. modo de color	4
3. estrategia	4
4. composiciones monocromáticas	5
5. composiciones de múltiples hues	6
6. composiciones clásicas	7
6.1. composiciones similares	7
6.2. composiciones complementarias	8
7. composiciones semánticas	9
8. references	11

combinando colores

1. el problema

hay casos en que por su naturaleza **es necesario tener muchos colores**. el caso clásico es un mapa, en el que muchos "países" limitan unos con otros y queremos que tengan colores distintos para **evidenciar claramente sus formas y límites**, como en la figura 1.

en algunos de estos casos **es imprescindible que sea el computador** (y no el diseñador) **el que define los colores** de una composición cromática. esto debido situaciones como una excesiva (o desconocida) cantidad de colores que se necesitan, lo que hace imposible decidir la composición cromática con antelación.

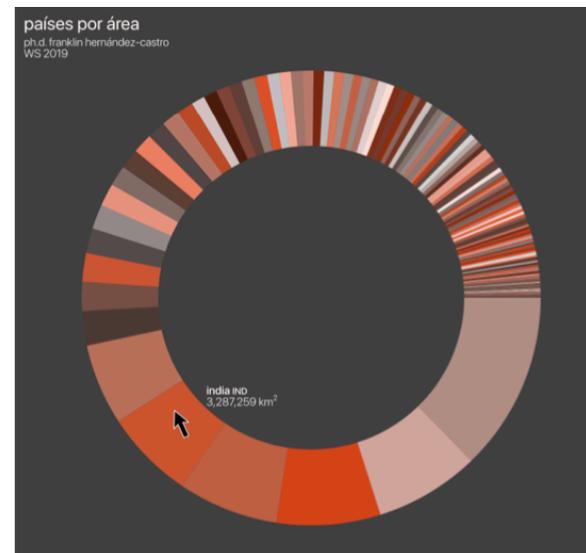
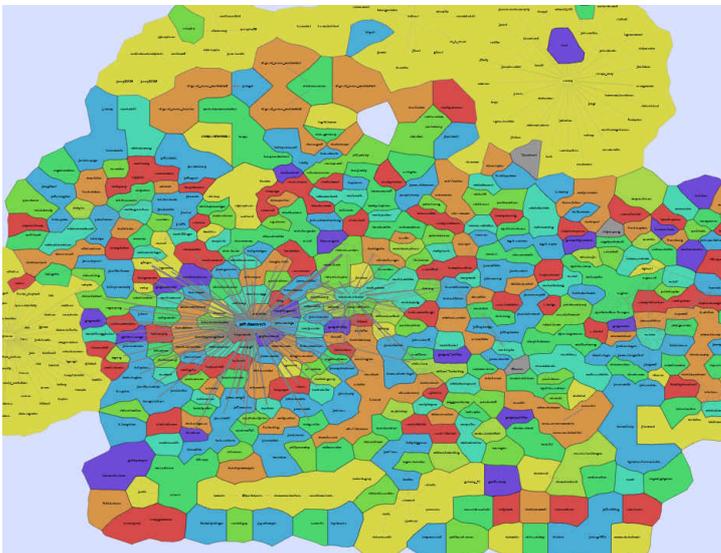


figura 1. caso típico en el que se usan muchos colores, a la derecha en un mapa (otto barca y beryn alvarado)¹ a la izquierda una visualización típica realizada automáticamente por código².

¹ ejemplo realizado por dos estudiantes de maestría (otto barca y beryn alvarado) en mi curso de visualización de datos para Intel®, donde se visualizan interacciones (en forma de correos electrónicos) en una corporación. los "países" son personas y su cercanía con otros depende de que tan a menudo se habían comunicado en un rango de tiempo específico.

² tomado de [2]

2. modo de color

lo primero que hay que decir es que en el mundo digital el **modo más usado** de manejo del color sería el **RGB** [3] [4]³. combinar colores armónicamente en **RGB es mucho más complejo** que en otros modelos de color. una vez más en "el error de le corbusier" [1] hay una amplia explicación de este punto específico.

para la combinación automática de colores, lo más cómodo es **pasarnos al modo de color HSB**, prácticamente en todos los lenguajes de programación tienen la posibilidad de manejar los colores desde este modelo.

el modelo (o modo) HSB, **divide las características del color en "hue"** o matiz, **saturación y luminosidad**. la estrategia detrás de manejar el color con HSB se basa en que: colores con características similares tendrán más probabilidades de ser armónicos que colores con características distintas. como este modelo nos permite manejar el matiz, la luminosidad y la saturación en forma independiente podemos controlar mejor la selección automatizada de colores "candidatos".

este modelo (o "modo", como se le dice en código "**color mode**") generalmente tiene escalas distintas. **el hue** se usa en una escala de **[0-360]** que recuerda que hablamos del "ecuador" del sólido de küppers [5].

por su parte, **la saturación y la luminosidad** se definen en **porcentajes** y por consiguiente usan una escala de **[0-100]**. ver figura 2.

sin embargo, cualquiera de estas escalas se pueden cambiar en muchos lenguajes de programación, pudiendo ponerlas en cualquier rango que se desee.

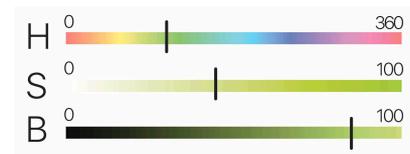


figura 2. sliders típicos de el modelo HSB con sus escalas más usadas

3. estrategia

para tener un enfoque sistemático en la combinación de colores en forma automática definiremos una regla:

"en un sistema HSB, para mantener una composición cromática armoniosa se debe mantener al menos uno de los ejes limitado a un rango pequeño o único"

³ si se desea más información sobre colores modos de color y escalas cromáticas ir a [3] y [4]

4. composiciones monocromáticas

ahora tratemos de aplicar la regla en los casos más comunes. el modo más usual y simple de usar colores profesionalmente es usar un solo color, de este modo la composición queda definida por las diferentes saturaciones y luminosidades de ese color.

nótese que para los diseñadores **los colores están definidos por su matiz** (o hue), o su posición en el ecuador de küppers [4][5], las otras **son variantes de ese color en saturación o luminosidad**, es decir latitud o profundidad en el sólido.

además **los grises, el blanco y el negro no son colores, son acromáticos**. este es el modo profesional de tratar con el color y es perfectamente coherente con el comportamiento de los canales RGB y el comportamiento de la luz misma.

aplicando nuestra regla podemos limitar el hue a un solo matiz y variar ampliamente los otros dos ejes S y B.

en la figura 3 vemos un diagrama de esto, los rectángulos en los ejes de saturación y luminosidad (H y B) muestran que queremos variar esos valores en todo el espectro, mientras que la línea en el eje del hue (H) muestra que deseamos usar un solo hue.

en código⁴ se vería más o menos así:

```
colors[i] = color(55, random(100), random(100));
```

en donde el matiz es 55, o sea **un número fijo**, mientras que la **saturación y la luminosidad es variable** en un rango de 0 a 100.

como es obvio, parto de que existe una matriz que se llama **colors [n]**, en la que se almacenan los **n** colores que se necesiten.

el resultado de esta estrategia son composiciones como las que se observan en la figura 4, donde se usó H con 55 y 300 respectivamente.



figura 4. composiciones monocromáticas.

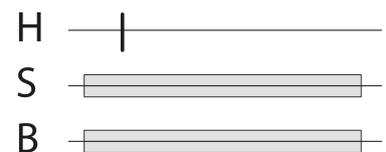
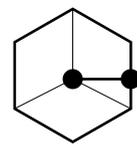


figura 3. diagrama mostrando una composición cromática de un solo hue y variando la saturación y la luminosidad en todo el espectro

⁴ usaré código en processing porque es bastante general

5. composiciones de múltiples hues

con esta metodología clara, probemos manejar mejor la combinación de colores logrando que sean armoniosos. nuestra regla dice:

"en un sistema HSB, para mantener una composición cromática armoniosa se debe mantener al menos uno de los ejes limitado a un rango pequeño o único"

en este caso vamos a mantener limitados dos ejes, el S y el B mientras nos movemos ampliamente por el eje H, sería como el contrario de la prueba anterior.

la idea general entonces es variar los matices (pues los necesitamos para algo como el mapa) pero **mantener las otras características** del color (como la luminosidad y la saturación) **similares** limitadas a rangos estrechos. por ejemplo, con un esquema como el de la figura 5.

en este caso, en los sliders, queremos decir que los matices (H) se moverán en todo su rango pero la saturación de los colores será baja, todos entre 0 y 50, y la luminosidad en cambio será alta, siempre entre 50 y 100.

en código la idea se vería algo así:

```
colors[i] =
color(random(360), random(50), random(50,100));
```

lo que genera algo así como colores "pastilla", como los que se muestran en la figura 6, luminosos pero poco saturados.

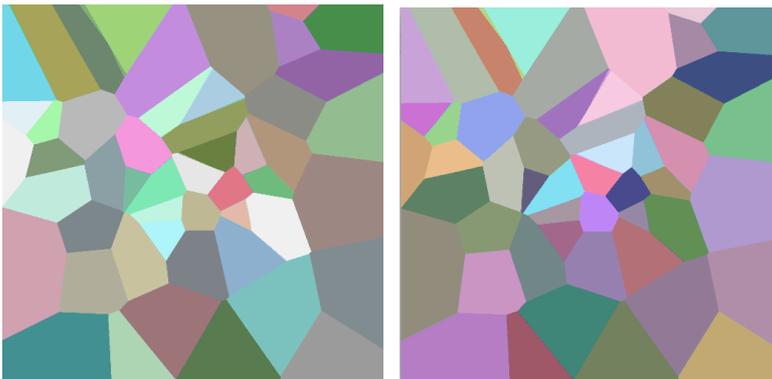


figura 6. composiciones resultado del código propuesto.

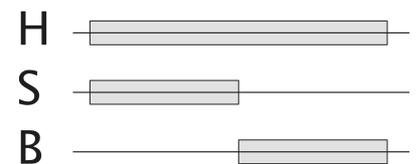


figura 5. diagrama mostrando una composición cromática con todo el espectro de hues y limitando la saturación y luminosidad

6. composiciones clásicas

con esta misma regla podemos trabajar las composiciones cromáticas clásicas como: "un solo hue", "similares", "complementarias"⁵, etc. [3]

las composiciones "de un solo hue" ya las hemos visto en el apartado 4 de composiciones monocromáticas.

6.1. composiciones similares

en el caso de composiciones similares podríamos probar limitar los tres ejes, lo que resultaría en algo así:

```
colors[i] =
color(random(170,200), random(50), random(50, 100));
```

lo que resultaría en composiciones como las de la figura 7.

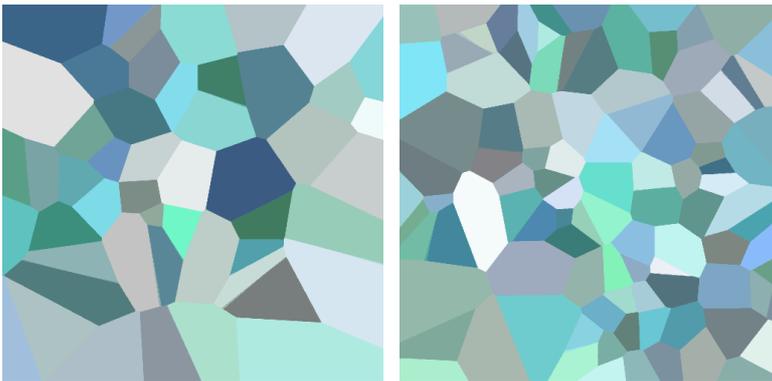


figura 7. composiciones similares, resultado del código propuesto.

o podemos usar la misma estrategia de "composición similar" pero con un rango distinto en hue:

```
colors[i] =
color(random(90), random(50), random(50,100));
```

y obtendríamos algo similar en otra temperatura cromática, fig. 8.

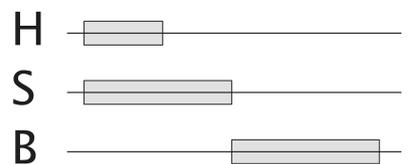
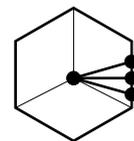
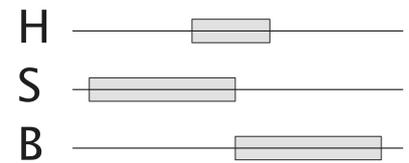
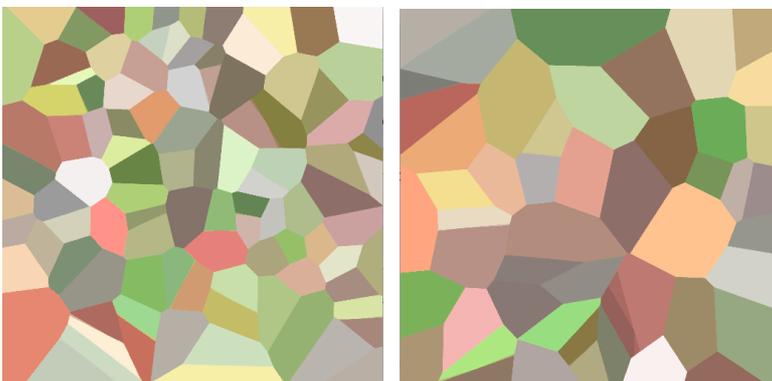


figura 8. composiciones similares, resultado del código propuesto.

⁵ todos estos aspectos están explicados con más detalle en [3]

6.2. composiciones complementarias

para colores complementarios debemos pensar en algo un poco más sofisticado. en este caso la limitación del eje H es en dos sectores del mismo, del lado rojo y del lado verde, pero solo de un matiz de cada lado, así que podríamos usar algo así:

```
if (random(1)>0.05) //dominancia de rojos 95%
  colors[i] = color(0, random(50), random(20,90));

else // acentos verdes 5%
  colors[i] =
  color(180, random(100), random(20,90));

if (random(1)>0.5) // 50% de acromáticos
  colors[i] = color(0, 0, random(20,90));
```

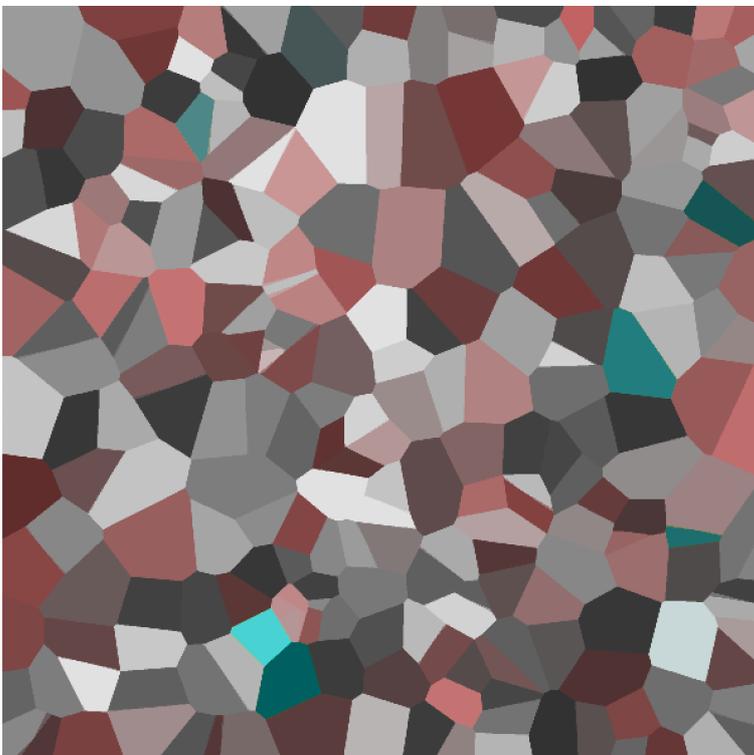
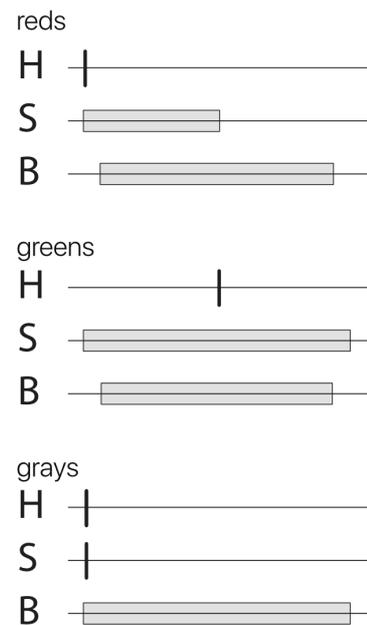


figura 9. composición complementaria, resultado del código propuesto.

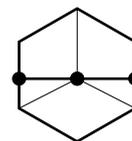


en la figura 9 (arriba), se muestra uno de los resultados del código propuesto. como se ve, el código divide los colores en tres grupos:

1. el 45% de los colores serán rojos con baja saturación y luminosidad amplia (20-90)
2. el 5% son verde con saturación alta
3. 50% son grises o acromáticos

el efecto es una composición de complementarios armónica, en este caso, rojo-verde. nótese que cuando la saturación es 0 la posición del hue resulta irrelevante pues hablamos de la línea entre el polo norte y el polo sur de romboedro [3], es decir los acromáticos.

de nuevo nuestra regla funciona aun en un limitación. "quebrada" del eje H.



7. composiciones semánticas

siguiendo esta simple regla se pueden controlar las composiciones cromáticas en la dirección que se desee, tratando de manejar **el clima cromático o la frase semántica** de la composición.

la selección de los rangos depende entonces de la semántica deseada en la composición. la metodología seguirá siendo controlable y reproducible en tiempo real si se mantiene la estrategia de que **las variables de HSB cambien en rangos angostos al menos en un eje.**

por ejemplo, si se desean **colores oscuros** pero muy saturados, se puede usar una base del luminosidad de 10% a 50% (10% para que no quedaran demasiados oscuros) y una saturación alta (digamos 50-100), lo que nos daría como resultado una paleta rica en hues pero siempre oscura y "misteriosa".

```
colors[i] =
color(random(360),random(50) + 50,random(10,50));
```

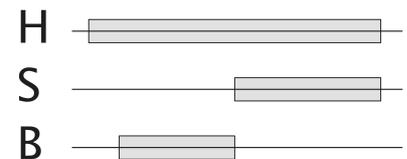
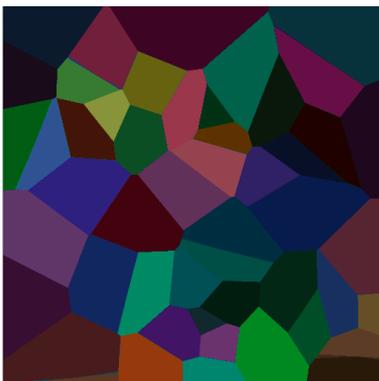


figura 10. composiciones con contenido semántico

por el contrario, la figura 11 muestra la misma composición pero con colores poco saturados. como se ve, la semántica asociada es completamente distinta.

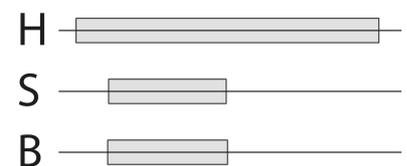
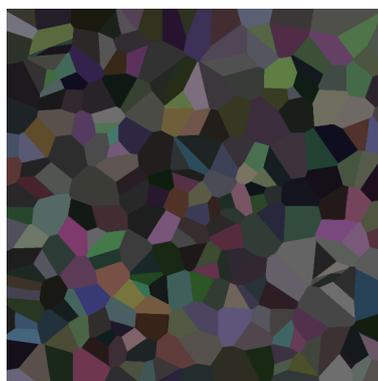
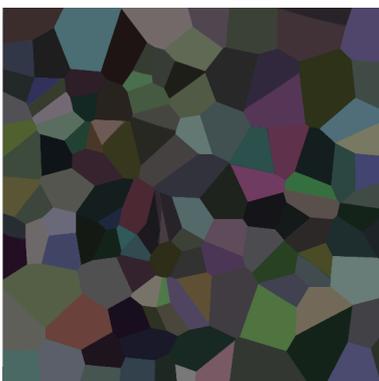


figura 11. composiciones resultado del código propuesto

así que “la semántica deseada” o “necesaria para el diseño” es lo que en última instancia define la composición. por ejemplo, en el caso de querer **colores pastel**, la estrategia podría ser esta:

```
colors[i] =
color(random(360), random(40), random(90,100));
```

cómo se ve, las saturaciones bajas y las luminosidades altas, condiciones comunes a todos los colores pastel, conservando la armonía general de la composición aun cuando los matices pertenecen a todo el espectro.

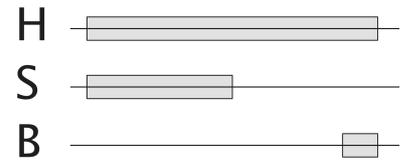
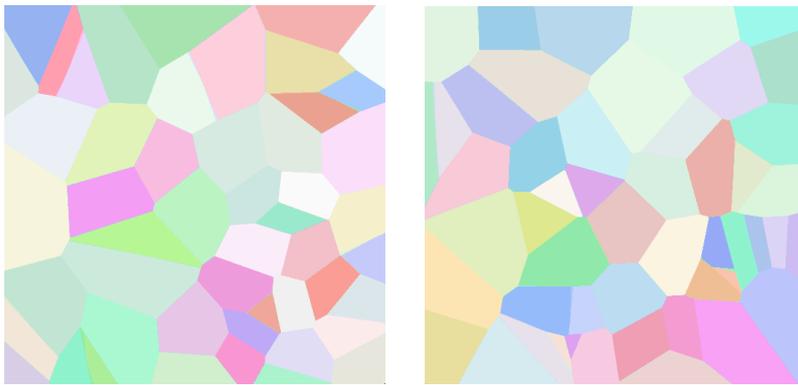


figura 10. composiciones resultado del código propuesto

volviendo a nuestro problema inicial (apartado 1), colorear un mapa desde código. ahora con nuestra regla es fácil generar un mapa con una composición cromática armónica, y con un número aleatorio de “países” o colores. problema solucionado!

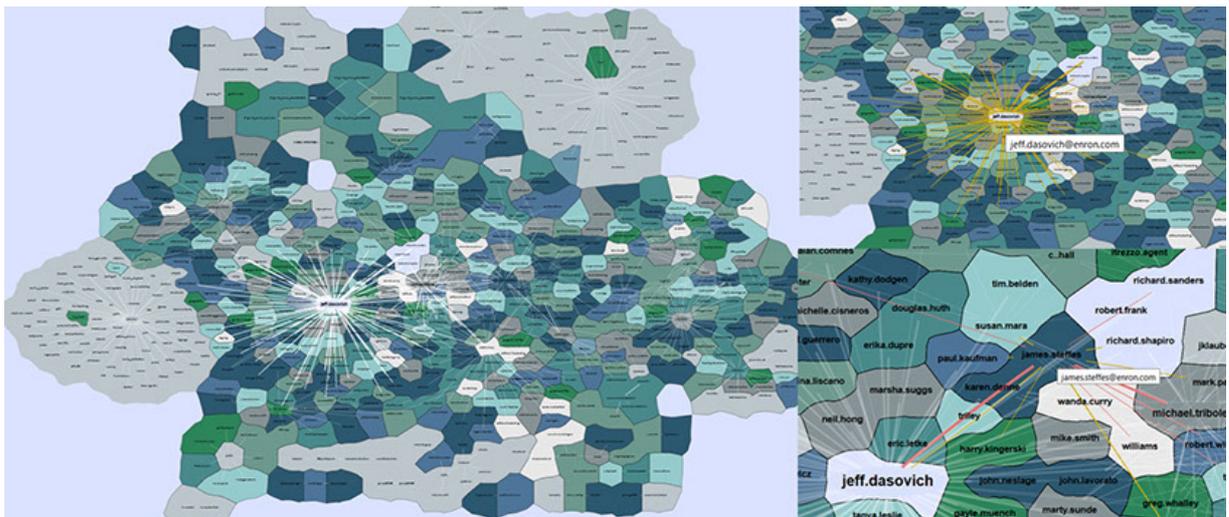
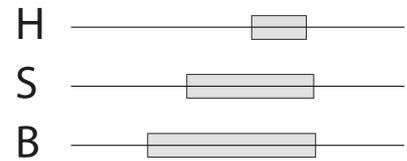


figura 11. choropleth (otto barca y berny alvarado)

8. references

- [1] Hernández-Castro, F. (2008). **El error de Le Corbusier**. Editorial Tecnológica de Costa Rica. Cartago, Costa Rica.
- [2] Hernández-Castro, F. (2021). **Dashboard design cookbook: metodología para el diseño de visualizaciones de datos para la toma de decisiones**. Instituto Tecnológico de Costa Rica. Repositorio TEC: <https://hdl.handle.net/2238/13281>
- [3] Hernández-Castro, F. (2016). **Ingredients: Teoría del Color**. Instituto Tecnológico de Costa Rica. Repositorio TEC: <https://repositoriotec.tec.ac.cr/handle/2238/7314>
- [4] Hernández-Castro, F. (2019). **cooking skills: diseñando escalas cromáticas por código**. Escuela de Diseño Industrial, Instituto Tecnológico de Costa Rica. Cartago, Costa Rica. Repositorio TEC: <https://repositoriotec.tec.ac.cr/handle/2238/10362>
- [5] Kueppers, H. (1980-2002). **Fundamentos de la teoría de los colores**. Ediciones Gustavo Pili, S.A. Barcelona.