

tf g

memoria

bellas artes  
2018-2019



**Mención:** Artes Visuales y Diseño

**Título:** Big Data pArt

**Estudiante:** Roc Gomar Calatayud

**Director:** Iván Albalate Gauchía

**PALABRAS CLAVE:**

**ALGORITMO, BIG DATA, ARTE, TRANSMEDIA, DEPORTE.**

**RESUMEN:** El siguiente trabajo consiste en una obra artística que se genera a partir de la aplicación de un algoritmo matemático, el cual a través de un proceso informático genera de forma mecánica obras de arte con un resultado plástico aleatorio. Para ello se hará uso de datos generados por personas, compartidos a través de las redes sociales dedicadas al deporte. Serán actividades físicas en las que para su realización requiera de un desplazamiento geográfico, las variaciones propias de ese movimiento servirán de parámetros que condicionaran las obras en todas sus dimensiones plásticas.





**ÍNDICE**

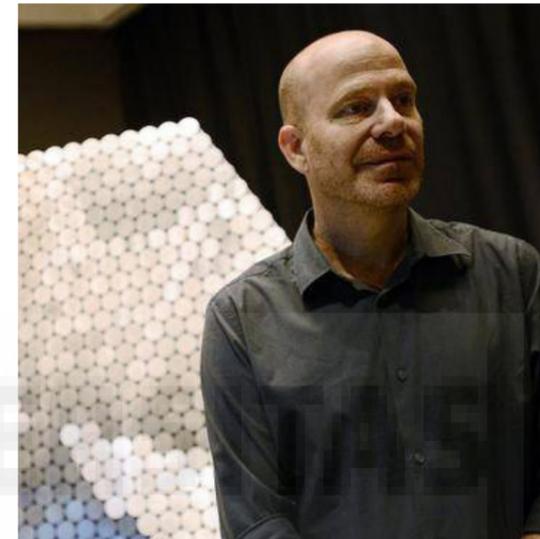
<b>1. Propuesta y objetivos</b>	<b>6</b>
<b>2. Referentes</b>	<b>8</b>
<b>3. Justificación de la propuesta</b>	<b>12</b>
<b>4. Proceso de producción</b>	<b>16</b>
<b>5. Resultados</b>	<b>24</b>
<b>6. Bibliografía</b>	<b>32</b>

## 1. Propuesta y objetivos



1. Mostrar arte digital generado por la combinación de esfuerzo humano e inteligencia artificial.
2. Diferenciar entre la obra generada por distintas personas aplicando un algoritmo.
3. Programar un algoritmo con la finalidad de producir obra artística.
4. Combinar la informática y el deporte como medio de generación de obra artística.
5. Generar la sensación de estar aventurado en la sala de exposición.
6. Reutilizar material informático obsoleto para la reproducción de las obras.

## 2. Referentes

**Daniel Rozin**

Es un artista que utiliza una técnica digital interactiva, en la que en su obra siempre está involucrado el espectador. Sus obras siempre tienen un componente mecánico que reacciona con el espectador. Para esto utiliza normalmente una cámara y un sensor de infrarrojos, donde un algoritmo lee la imagen y movimientos del espectador para reproducirla a forma de espejo mecánico de distintos materiales.

De este artista, nos ha influenciado, a parte de que se trata una obra digital, que la obra la va realizando un algoritmo sobre los datos que le envía la cámara y los infrarrojos.

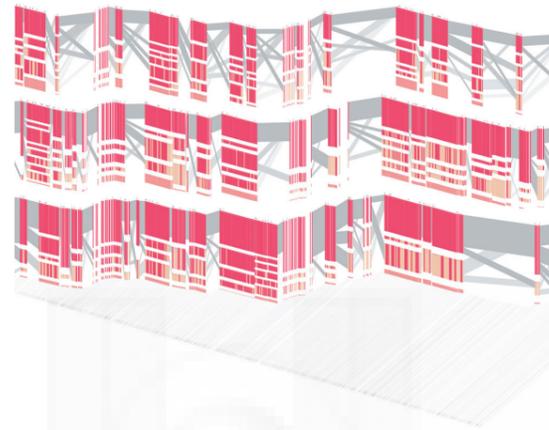
**David Cuartielles**

Es un ingeniero/artista creador junto a Massimo Banzi de *Arduino*, una placa programable de hardware libre, que en sus inicios fue creada por sus inquietudes artísticas, aunque después ha servido para muchos ámbitos más que para el arte. Para la programación de arduino se basó en el IDE de programación de *Processing*, herramienta que se ha utilizado para crear este proyecto artístico



**Ben Fry**

Es el co-creador de *Processing*, experto en visualización de datos. Ben creó el *Processing*, como el lenguaje para el arte, siendo un lenguaje de programación destinado a ofrecer visualizaciones gráficas de forma en que el aprendizaje de este no es lo complicado que puede ser cualquier otro lenguaje. Cualquier artista con un poco de esfuerzo puede aprenderlo y realizar sus obras. Es la herramienta que se a utilizado para mostrar los datos leídos en forma de imagen digital.

**Nathalie Miebach**

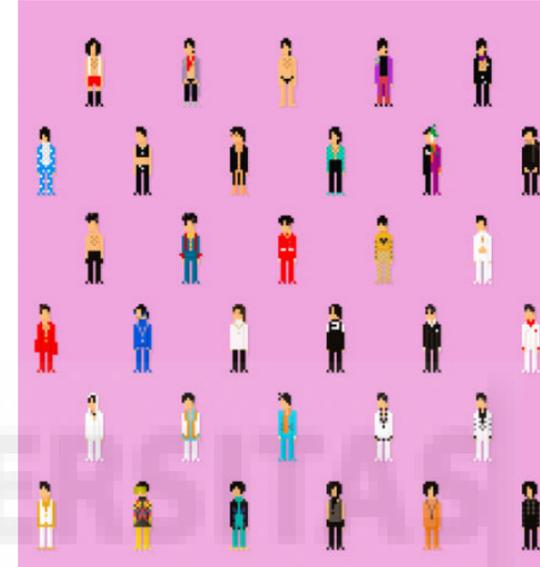
Centra su trabajo en la intersección del arte, la ciencia y la articulación visual de las observaciones científicas. Utilizando metodologías y procesos de ambas disciplinas, traduciendo estos datos científicos relacionados con la ecología, el cambio climático y la meteorología en estructuras tridimensionales para generar su obra artística.

Esta forma de trabajar a influido mucho en la realización de la obra, ya que este trabajo analiza una serie de datos que genera una persona, para interpretarlos como una imagen gráfica.

**Craig Robinson**

Ha retratado las caras más conocidas del cine, la música y la cultura popular a golpe de pixel. Es capaz de reducir a un puñado de cuadrados de colores la esencia de una persona y mostrar en un espacio reducido sus características más representativas.

Este artista nos ha indicado el camino a seguir para obtener el resultado final de la imagen, sintetizar varios datos en un solo pixel.



UNIVERSITAT DE VALÈNCIA  
Miguel  
Hernández

### 3. Justificación de la propuesta

Podemos considerar que en cualquier manifestación artística existen tres factores principales que influyen en su generación, en primer lugar tenemos el medio con el que se realiza, como puede ser medios líquido propios de la pintura, materiales sólidos en la escultura, o medios sutiles en el arte digital, por ello la elección del medio seguramente decantara la disciplina artística utilizada o combinación de las mismas. En segundo lugar, nos encontramos con la herramienta en sí misma que ejecuta la pieza, por ejemplo un formón para tallar una escultura, una pistola de aire caliente para formar una pieza con plástico, o los utensilios pictóricos para generar un cuadro, de este modo dependiendo del medio necesitaremos unas herramientas u otras. En tercer y último lugar encontramos al ejecutor de dicho proceso, al artista, el individuo o colectivo encargado de pensar, procesar, programar y llevar a cabo mediante la utilización de las herramientas la transformación del medio en una pieza artística.

A lo largo de la historia, el arte ha ido evolucionando multiplicando las disciplinas los medios y las herramientas, pasando de las primeras pinturas rupestres al actual videoarte, performance, arquitectura ficcional... todo esto gracias también a los cambios que han propuesto los artistas que han ido evolucionando y adaptándose, conjuntamente con la sociedad la época en la que se encuentran.

Actualmente podemos decir que estamos en una época en la que el deporte ha adquirido una relevancia notable, debido a que en nuestra sociedad, en el ambiente en el que nos encontramos, la gente ha pasado de la ruta del bakalao al ciclismo en ruta, running o alguna de las nuevas disciplinas de deporte que se han creado. Antes la gente optaba por las drogas para aguantar mejor la fiesta y ahora optan por las mismas, pero, para rendir más en el deporte y ser capaces de estar el máximo tiempo posible en sufrimiento.

Por otra parte, nos encontramos inmerso en la sociedad de la información, del big data. En cualquier faceta de nuestra vida generamos datos que son almacenados en algún medio informático. La mayoría de la gente utiliza algún tipo de red social, no se para de recopilar imágenes en formato digital, en nuestros trámites burocráticos todo está informatizado. Tenemos verdaderos vertederos de datos. Donde todo el mundo quiere tener privacidad haciendo pública su vida.

¿Un esfuerzo atlético, se puede convertir en una obra de arte? Si. Si medimos nuestras constantes vitales mientras hacemos deporte, tenemos el medio para crear la obra, como si de una roca de mármol se tratara, nos falta la herramienta. Como herramienta se va a utilizar el tratamiento de los datos que producimos al hacer deporte. Y ¿quién se encarga de dar forma a esto?, la respuesta la encontramos sencilla, un algoritmo.

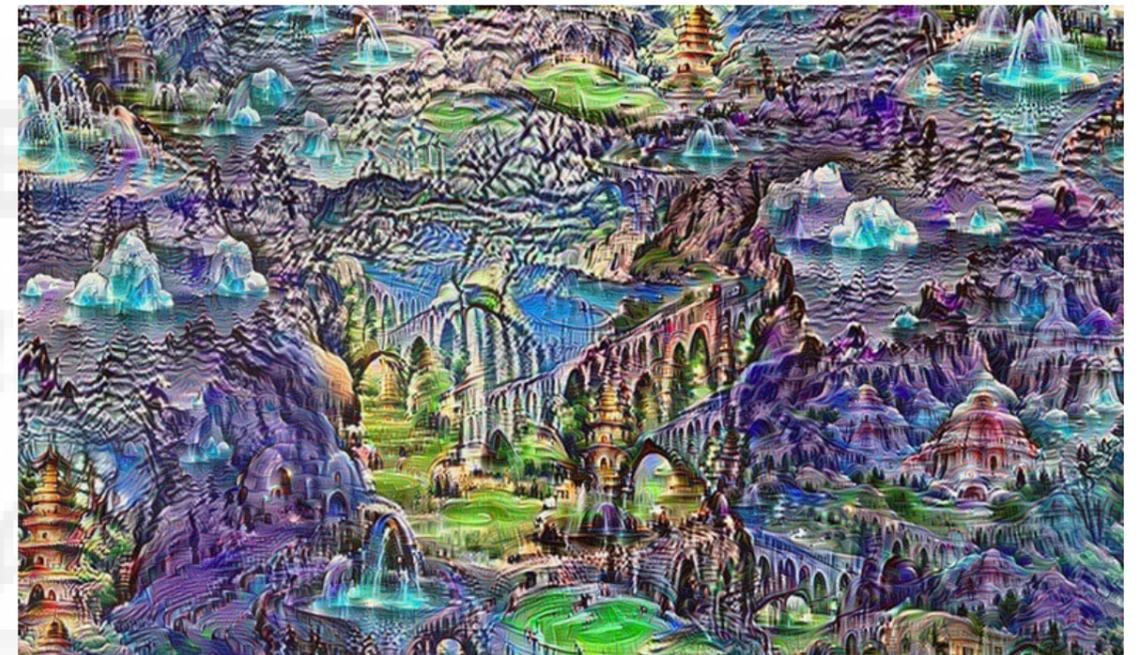
Con esto, nuestra intención es la de cambiar el triángulo del proceso artístico, modificando el artista por un algoritmo con inteligencia artificial.

Podemos decir, que incluir algoritmos en el mundo del arte es una tendencia que cada vez más va progresando, son muchos los artistas que se dedican a programar el algoritmo que luego se encargará de generar el arte, como puede ser David Cuartielles, con el uso de Arduino y Processing, para que mediante el medio que en este caso son las personas que van a la exposición, el algoritmo reaccione de una forma u otra haciendo al espectador participe de la obra. Otro caso es el de Daniel Rozin, que a sorprendido con el uso de “microrobots” en sus obras, que comandados por un algoritmo son capaces de interactuar con el espectador.

Otra finalidad es la de utilizar software y hardware libre en la medida de lo posible. El software va a ser 100% libre, ya que se va a realizar en Processing, Python y para la reproducción los equipos contarán con sistema Linux o Android. Para la reproducción se utilizarán equipos de hardware libre como es la Raspberrypi, pero también se pretende re-utilizar equipos obsoletos o desechados.

Como indicamos anteriormente, para la reproducción de los diversos resultados pictórico-digitales que nos proporcionará el algoritmo, se va hacer con material informático y/o audiovisual reciclado, como son equipos obsoletos y/o estropeados (a nivel lógico). Adaptándose para correcta reproducción audiovisual de las distintas reproducciones. La finalidad de este uso, a parte de la económica es la de dar vida todavía a equipos que la gente por diversas razones se ha deshecho de ellos, demostrando que todo puede tener un uso distinto para el que ha sido concebido y es útil, mientras no este roto de verdad.

Para la ambientación de la exposición se buscará que esta se exponga en distintas habitaciones no muy grandes o pasillos más bien estrechos, donde se dispondrán distintos tipos de pantallas colgadas a distintas alturas, donde se verá la obra pictórico-digital y se escuchara el ritmo del corazón en cada una de las reproducciones, en la pared de enfrente se encontrará una imagen del mapa topográfico de la zona donde el deportista ha realizado la actividad. En caso de ser un espacio diáfano, se puede construir este escenario “estrecho” con paneles móviles, formando una especie de laberinto abierto.



*The Next Rembrandt - Microsoft.*

La finalidad de esta obra es la de traducir de forma gráfica, los datos que genera un atleta cuando está realizando un esfuerzo físico deportivo.

Para esto entre varias opciones, como *Polar My Trainer*, *Garmin Connect*, *Suunto*... después de analizar las principales, se tomó la decisión de utilizar *Strava*, ya que es una red social de deportistas, donde la gente cuelga los datos de sus entrenamientos y competiciones.

#### 4. Proceso de producción



Para esto utilizamos los archivos propios y de voluntarios que tienen esta red social, archivos en formato .gpx que son un tipo de xml de geolocalización, donde se guardan multitud de datos, en este caso se ha utilizado longitud, latitud, frecuencia cardiaca, elevación y temperatura.

Para poder leer y extraer estos datos se ha hecho una función con *Python*, un lenguaje de programación de código abierto que ahora está en auge por sus altas capacidades y funcionalidades, los gpx, son archivos complejos, que dependiendo del tipo de datos que se quiera extraer se ha de utilizar una técnica u otra distinta. Todos estos datos leídos se almacenan en una lista de diccionarios.

El algoritmo abre el archivo con permisos de lectura, y recorre sus miles de líneas guardando sólo los datos deseados y convirtiéndolo en el tipo de datos adecuado para que luego *Processing* los pueda interpretar.

```



```

Extracto de archivo .gpx.

```

def lecturagpx(arxiu):
    track = open(arxiu, "r")
    nom = track.name
    ruta = []
    return = []
    atemp = None
    hr = None
    lon = None
    lat = None
    ele = None
    ubicacions = 0
    topeLon = None
    topeLat = None
    minlat = None
    minlon = None
    llarg = len(track.readlines())
    track.seek(0)
    punts = track.readlines()
    for punt in punts:
        if punt.find("<gpstpx:atemp>")!=-1:
            tagtemp = match(punt, "<gpstpx:atemp>(.*?)</gpstpx:atemp>")
            atemp = int(tagtemp[1])
        elif punt.find("<gpstpx:hr>")!=-1:
            taghr = match(punt, "<gpstpx:hr>(.*?)</gpstpx:hr>")
            hr = int(taghr[1])
        elif punt.find("<trkpt>")!=-1:
            ll = matchAll(punt, "(.*?)")
            lat = float(ll[0][1])
            lon = float(ll[1][1])
            if ubicacions == 0:
                topeLat = lat
                topeLon = lon
                minlat = lat
                minlon = lon
                ubicacions+=1
            if lat > topeLat:
                topeLat = lat
            if lon > topeLon:
                topeLon = lon
            if lat < minlat:
                minlat = lat
            if lon < minlon:
                minlon = lon
        elif punt.find("<ele>")!=-1:
            tagele = match(punt, "<ele>(.*?)</ele>")
            ele = float(tagele[1])
        elif punt.find("<trkpt>")!=-1:
            ruta.append({'lat':lat,'lon':lon,'temp':atemp, 'hr':hr,'ele':ele})
    track.close()
    #print ruta
    print llarg
    print "cantitat de punts " ,ubicacions
    print topeLat, " " ,minlat
    print topeLon, " " ,minlon
    print topeLat-minlat
    print topeLon-minlon
    tamany = [int(round((topeLat-minlat)*10000,0)),int(round((topeLon-minlon)*10000,0))]
    print "tamany=", round((topeLat-minlat)*10000,-1), " " ,round((topeLon-minlon)*10000,-1)
    print tamany
    back = int(ubicacions/tamany[0])
    print back
    return = [tamany,ruta,topeLon,topeLat]
    return return

```

Funcion *Python* que lee el archivo .gpx.

```

from funciones import *
a = 0
b = 0
esquerra = 1
contador = 0
nom = "track2.gpx"
ruta = lecturagpx(nom)
ample = round(ruta[0][0],-1)
llarg = round(ruta[0][1],-1)
segons = len(ruta[1])
puntsx = []
puntsy = []
ptcolor = [0,0,0]
mitja = 0
diferencial = ((int(round(ample,-1))/10)*(int(round(llarg,-1))/10))/segons
print "diferencial ", diferencial
def setup():
    global puntsx, puntsy
    size(int(round(ample,-1)),int(round(llarg,-1)))
    colorMode(HSB)
    background(0)
    noStroke()
    frameRate(60)

for rut in ruta[1]:
    pointx = round(int(abs(rut.get("lat")-ruta[3])*10000),-1)
    pointy = round(int(abs(rut.get("lon")-ruta[2])*10000),-1)
    if (pointx,pointy) not in zip(puntsx,puntsy):
        puntsx.append(int(pointx))
        puntsy.append(int(pointy))

def draw():
    global diferencial
    global ample
    global llarg
    global ruta
    global llista
    global a
    global b
    global contador
    global esquerra
    global mitja
    global segons
    global nom
    if a <= ample-10 and a >= 0:
        rg = int((255*ruta[1][contador].get('hr'))/200)
        vr = int((255*ruta[1][contador].get('ele'))/2500)
        bl = int((255*ruta[1][contador].get('temp'))/48)

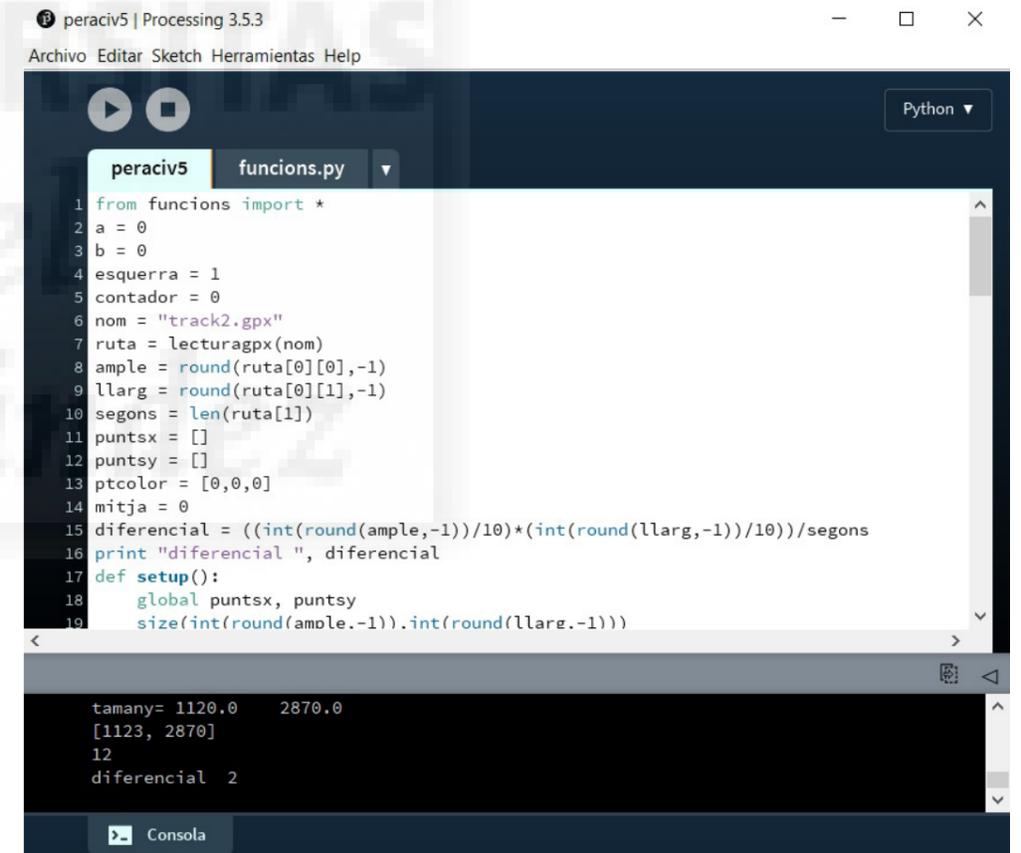
        fill(rg,vr,bl)
        if (a,b) not in zip(puntsx,puntsy):
            rect (a,b,10,10)
        else:
            fill(rg*5,vr*3,bl*4)
            rect (a,b,10,10)
        if esquerra == 1:
            a += 10
        else:
            a -= 10
            if contador < segons-1:
                contador += 1
            else:
                print "entra 1"
                contador = 0
    elif b < llarg-10:
        b += 10
        if esquerra == 1:
            a = ample-10
            esquerra = 0
        else:
            a = 0
            esquerra = 1
            if contador < segons-1:
                contador += 1
            else:
                print "entra 2"
                contador = 0
    else:
        print contador
        guardar = nom+".tif"
        save(guardar)
        noLoop()
        print a
        print b

```

Funcion Pocessing que interpreta y dibuja.

Una vez la función nos da ya los datos debidamente traducidos, *Processing* se encarga de analizar y procesar estos datos para dar un resultado gráfico, que se va generando, y una vez realizado guarda una imagen del estado final del análisis de forma gráfica. La frecuencia cardiaca se ha asociado al rojo, la altura al verde y la elevación al azul, cada segundo tenemos un dato nuevo que son los que forman en forma de píxel el fondo de la obra. La longitud y la latitud se han utilizado, para apagar los pixeles en el lienzo, representando la ubicación en cada momento del atleta.

Esta obra se debe exponer como se comenta en la justificación de la propuesta.



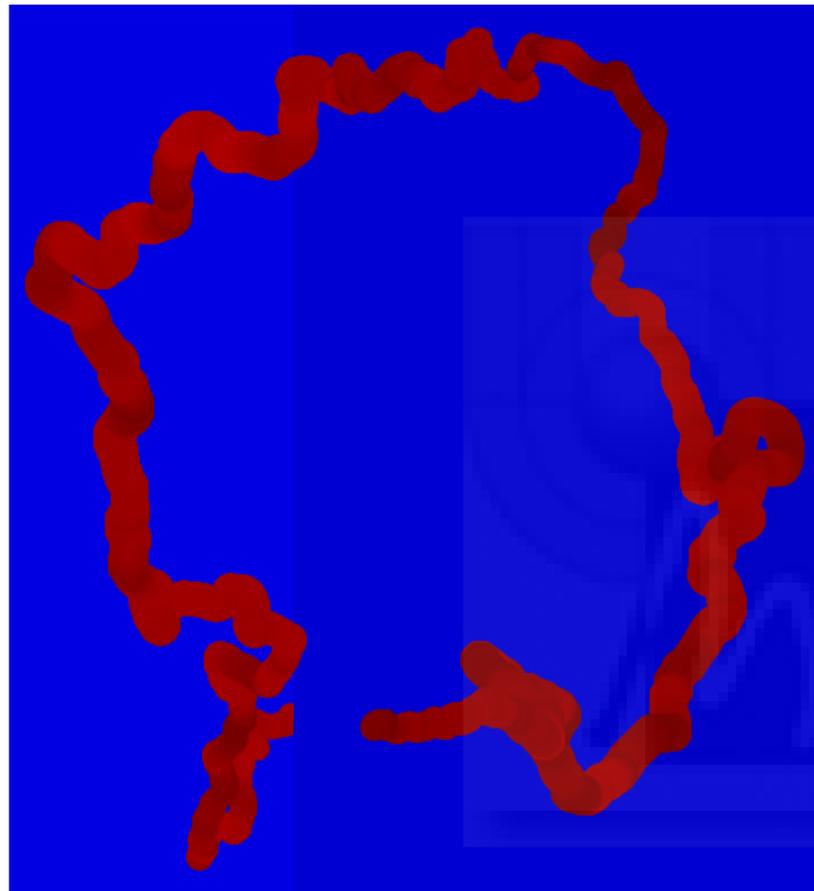
```

peraciv5 | Processing 3.5.3
Archivo Editar Sketch Herramientas Help
Python
peraciv5 funciones.py
1 from funciones import *
2 a = 0
3 b = 0
4 esquerra = 1
5 contador = 0
6 nom = "track2.gpx"
7 ruta = lecturagpx(nom)
8 ample = round(ruta[0][0],-1)
9 llarg = round(ruta[0][1],-1)
10 segons = len(ruta[1])
11 puntsx = []
12 puntsy = []
13 ptcolor = [0,0,0]
14 mitja = 0
15 diferencial = ((int(round(ample,-1))/10)*(int(round(llarg,-1))/10))/segons
16 print "diferencial ", diferencial
17 def setup():
18     global puntsx, puntsy
19     size(int(round(ample,-1)),int(round(llarg,-1)))
tamany= 1120.0 2870.0
[1123, 2870]
12
diferencial 2
Consola

```

Interfaz de Processing

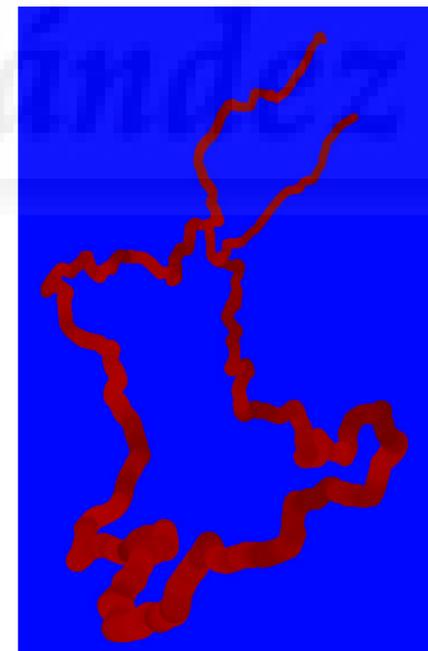
Los primeros resultados obtenidos, se representan en forma de video o imagen, como unas pinceladas que van cambiando su tamaño e intensidad de color, el fondo se mostraba los cambios de temperatura.



Primeros bocetos.



Primeros bocetos.

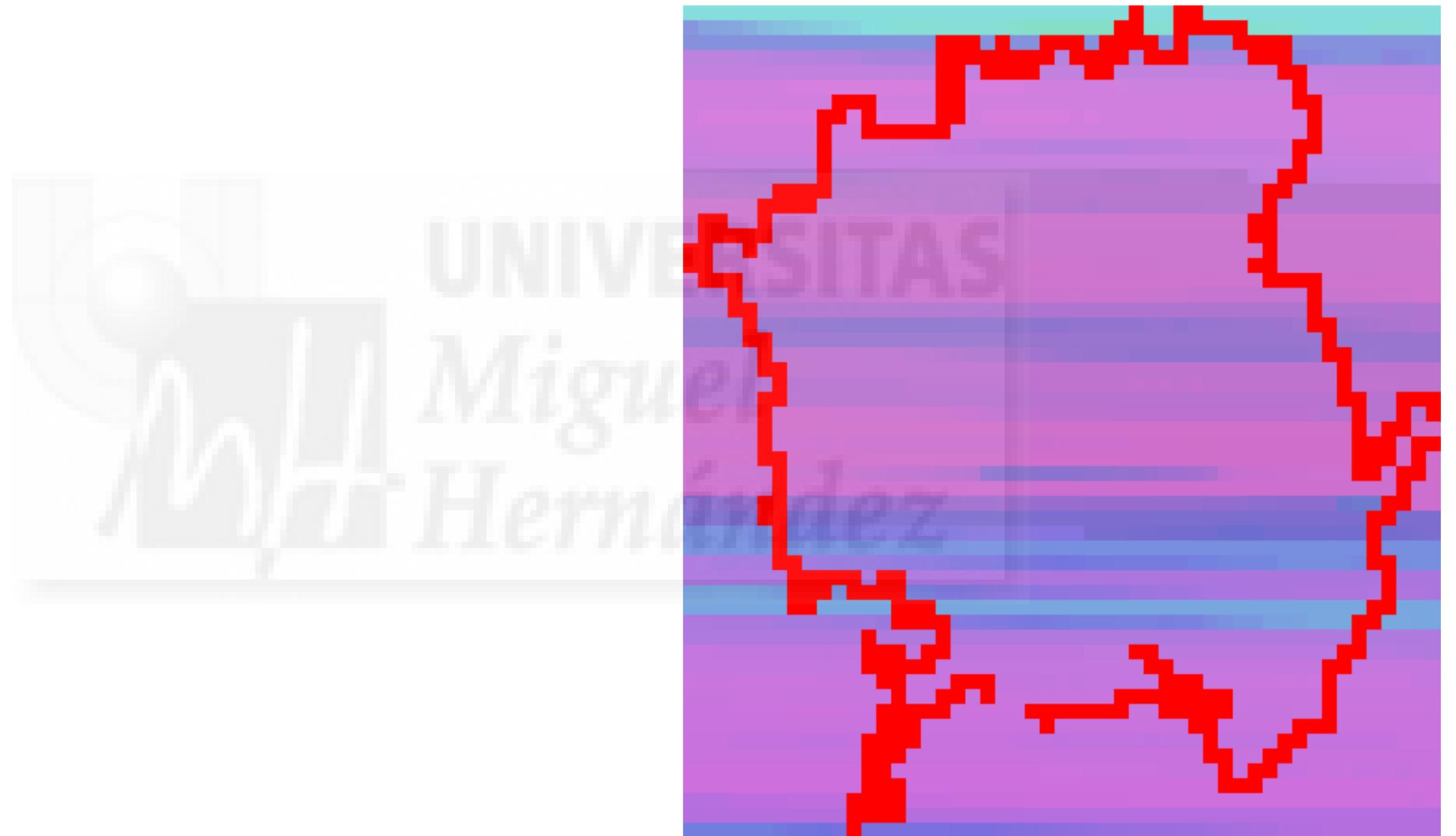


Primeros bocetos.

Debido a que en la mayoría de los casos las variaciones de temperatura eran mínimas, sólo en casos concretos podían dar variaciones notables de color, se decidió cambiar la forma de interpretación de los datos. Al mismo tiempo la representación de la ruta también se mostrará de forma más pixelada, no tan orgánica, para dar más sensación de algo hecho por una máquina.

En el resultado final se ha tratado de dar una técnica de PixelArt, tratando las imágenes como lo haría una computadora de 8-bits, de los años 80.

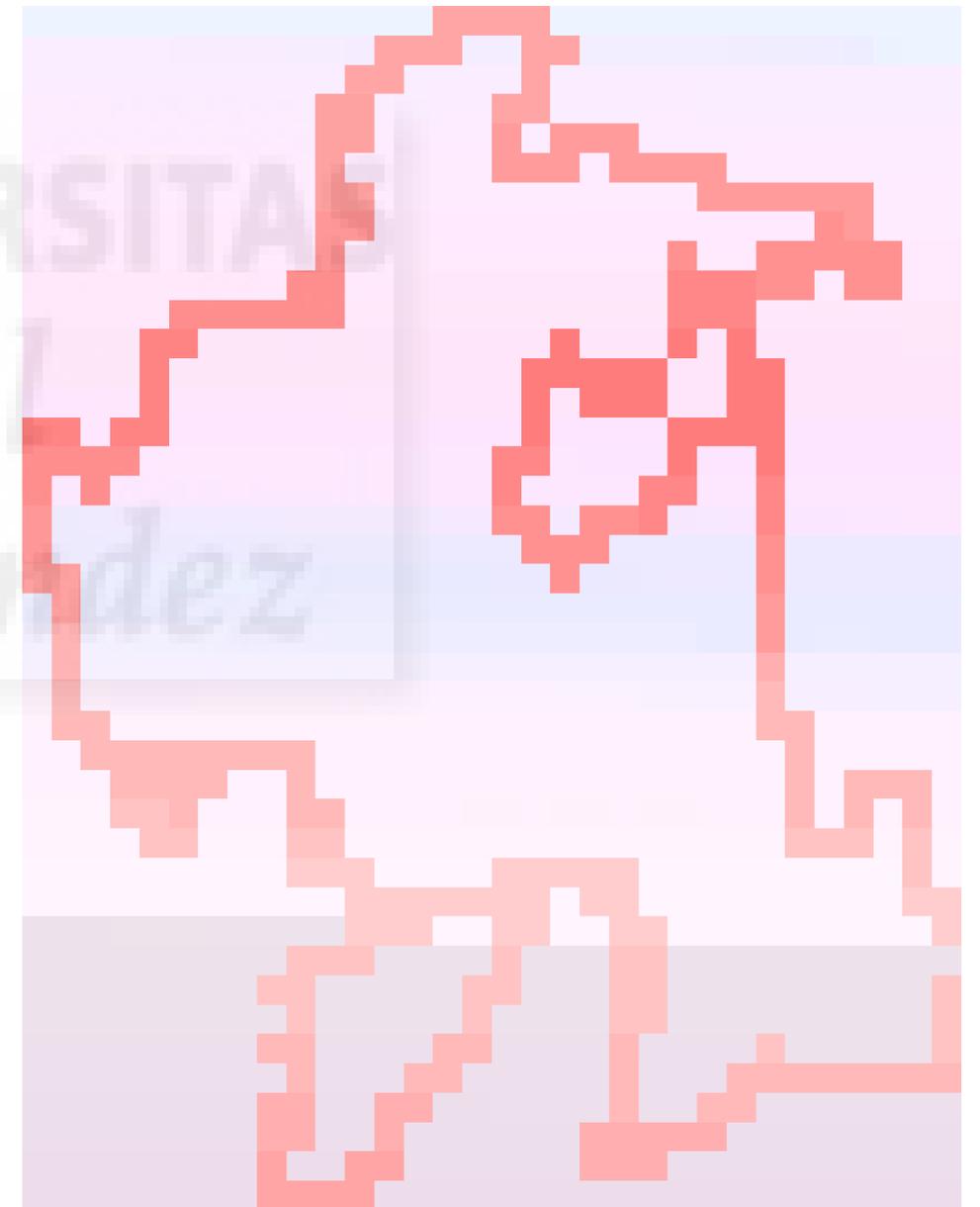
5. Resultados



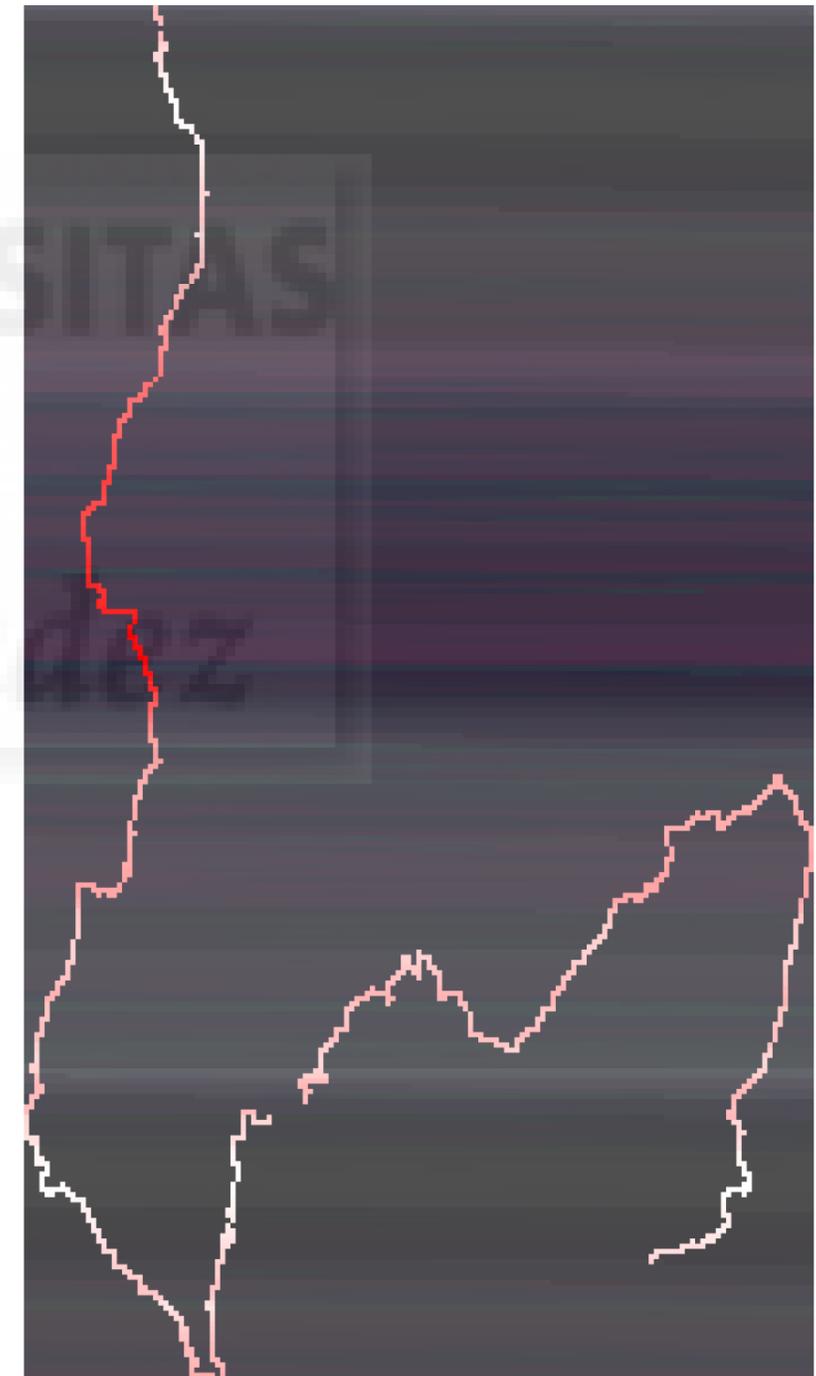
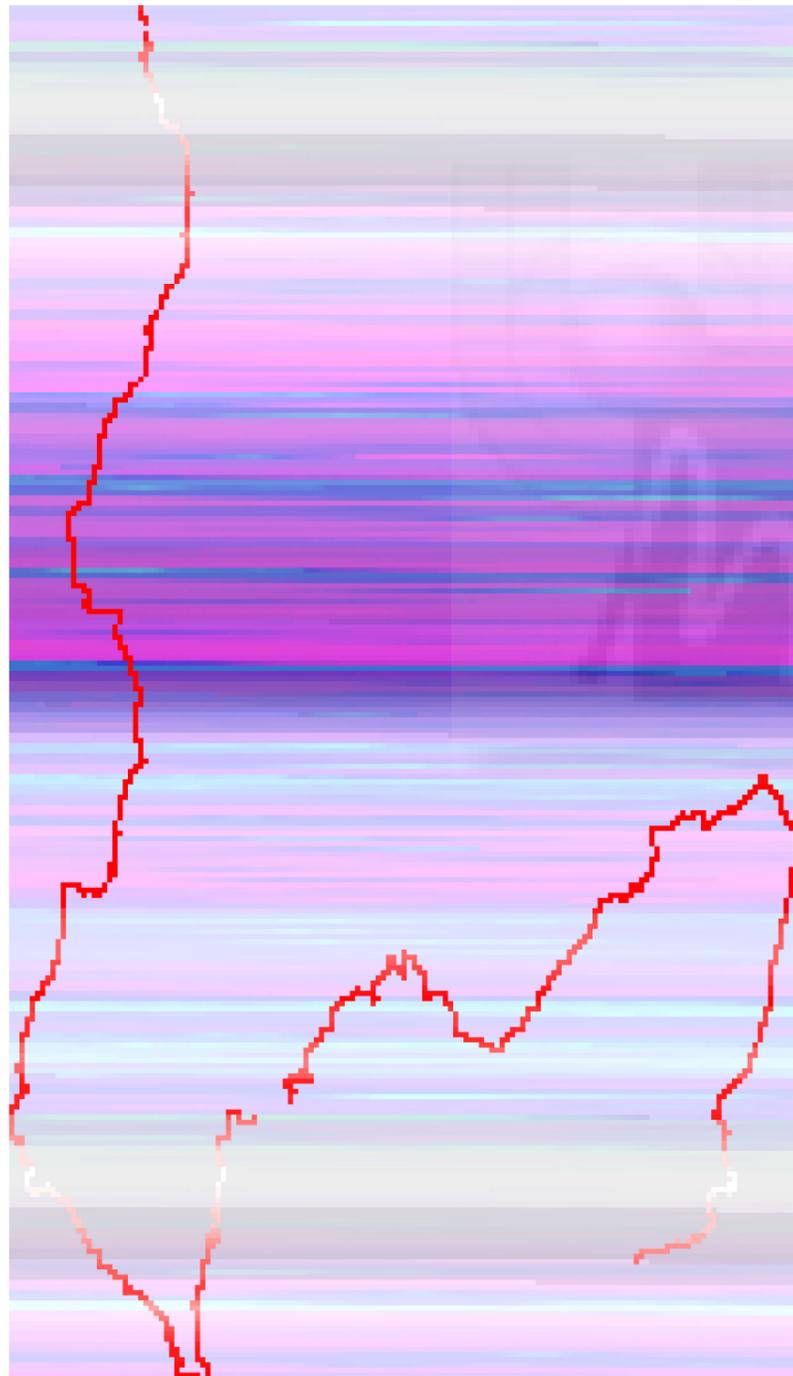
Misma ruta realizados en periodos distintos, se puede observar, como en la imagen inferior los colores son más nítidos, cosa que significa un esfuerzo más intenso, sin embargo en la imagen de la pagina siguiente, se observa como un color más claro, menos intensidad.



UNIVERSITAT  
*Miguel*  
Hernández

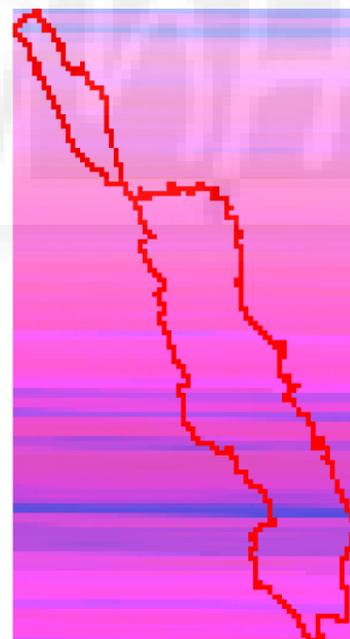
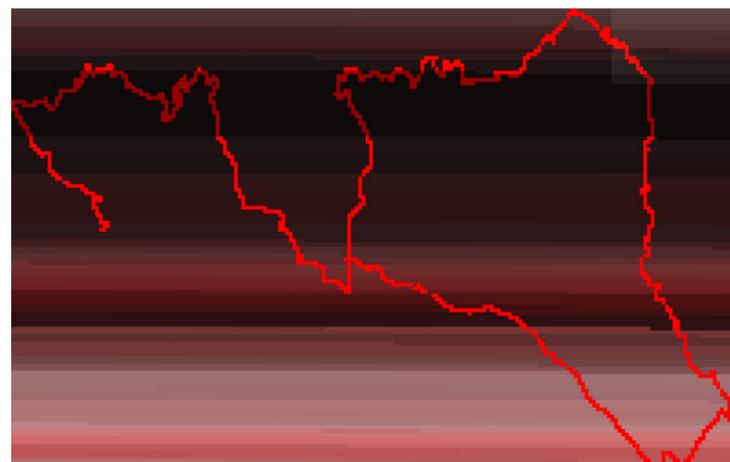


En las dos siguientes reproducciones, podemos observar la misma ruta, al mismo tiempo por dos atletas distintos, el que tiene tonalidades azules, su nivel de esfuerzo es tranquilo, el que aparece con tonalidades grises, llevaba un nivel de esfuerzo máximo durante toda la ruta. Esto nos demuestra la diferente condición física de los dos deportistas.





Ejemplo de exposición



Ejemplo de exposición

## 6. Bibliografia



- <https://py.processing.org/>
- <https://www.python.org/>
- <https://www.grandpalais.fr/es/node/17985>
- <https://www.ted.com>
- <https://www.arduino.cc/>
- <https://www.raspberrypi.org/>