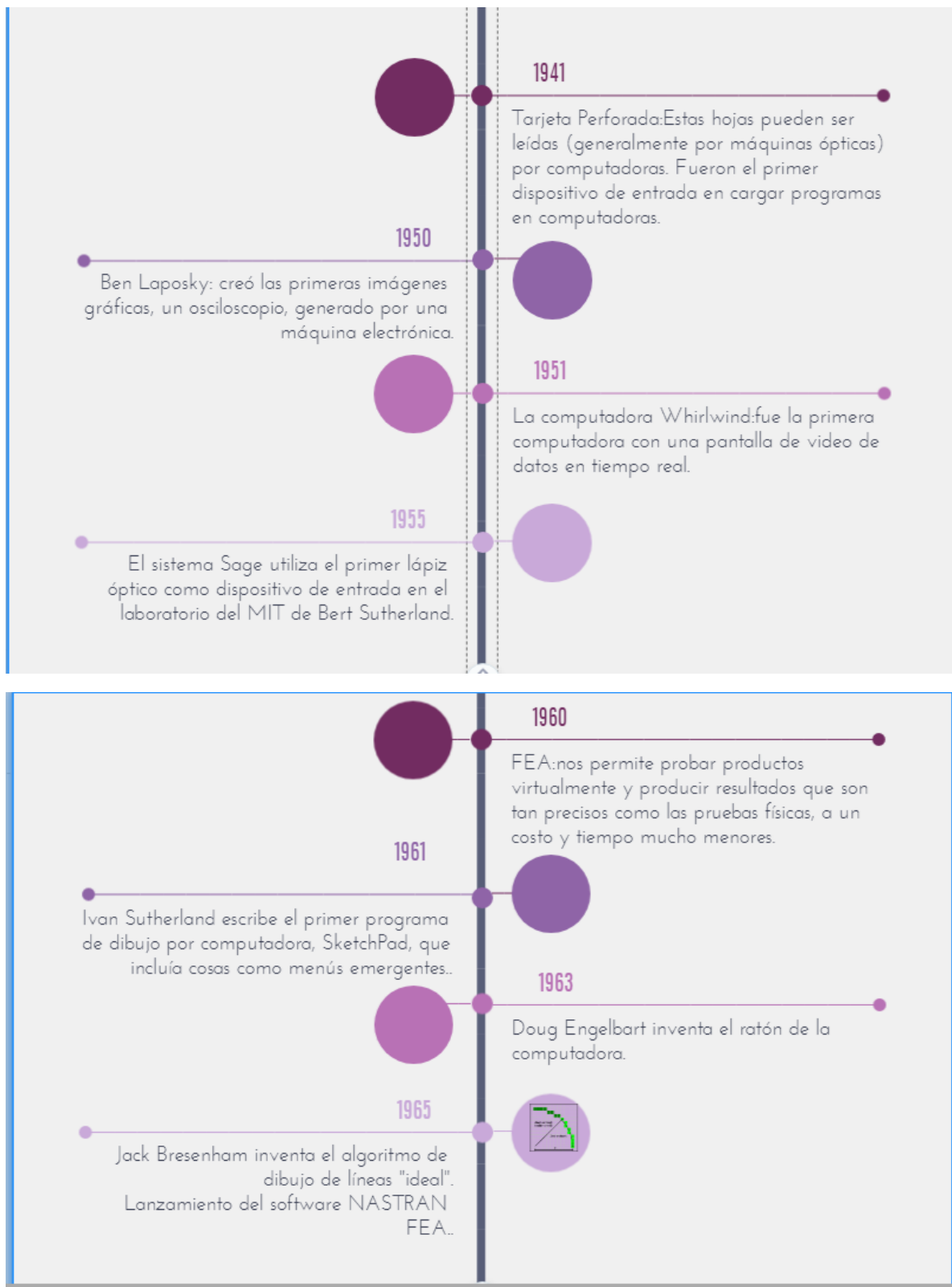
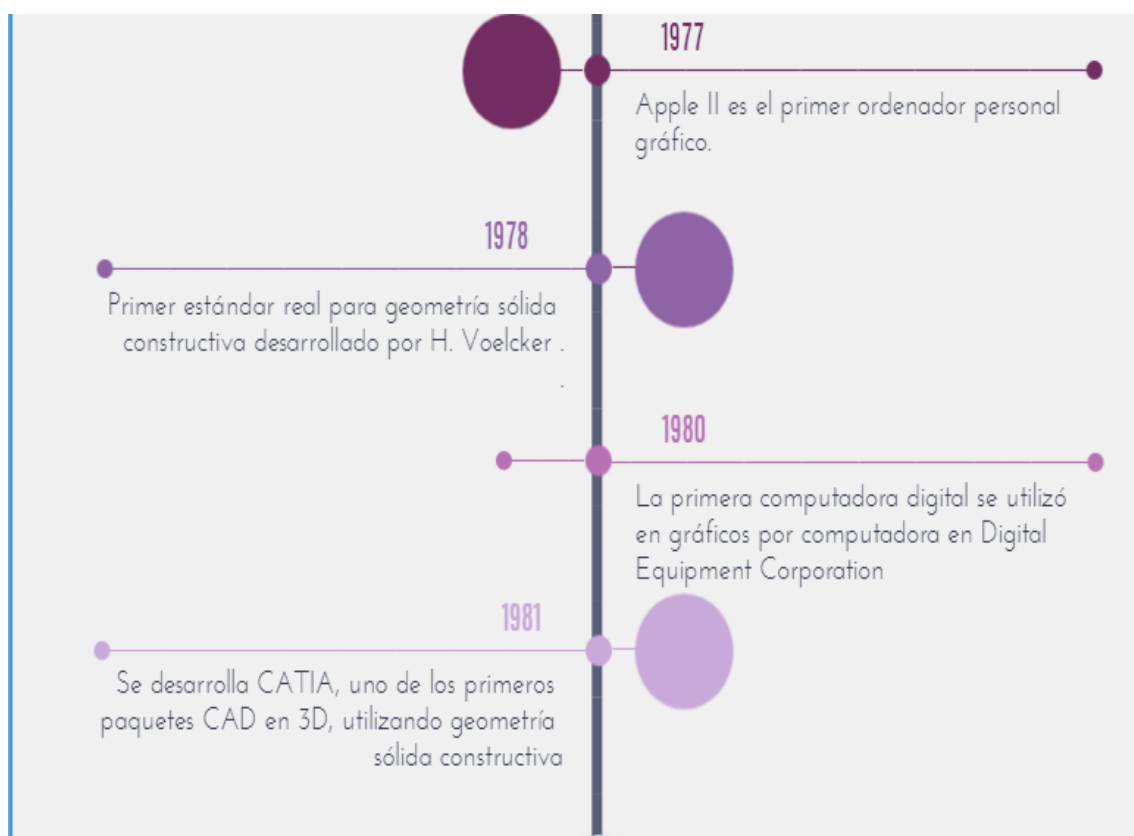
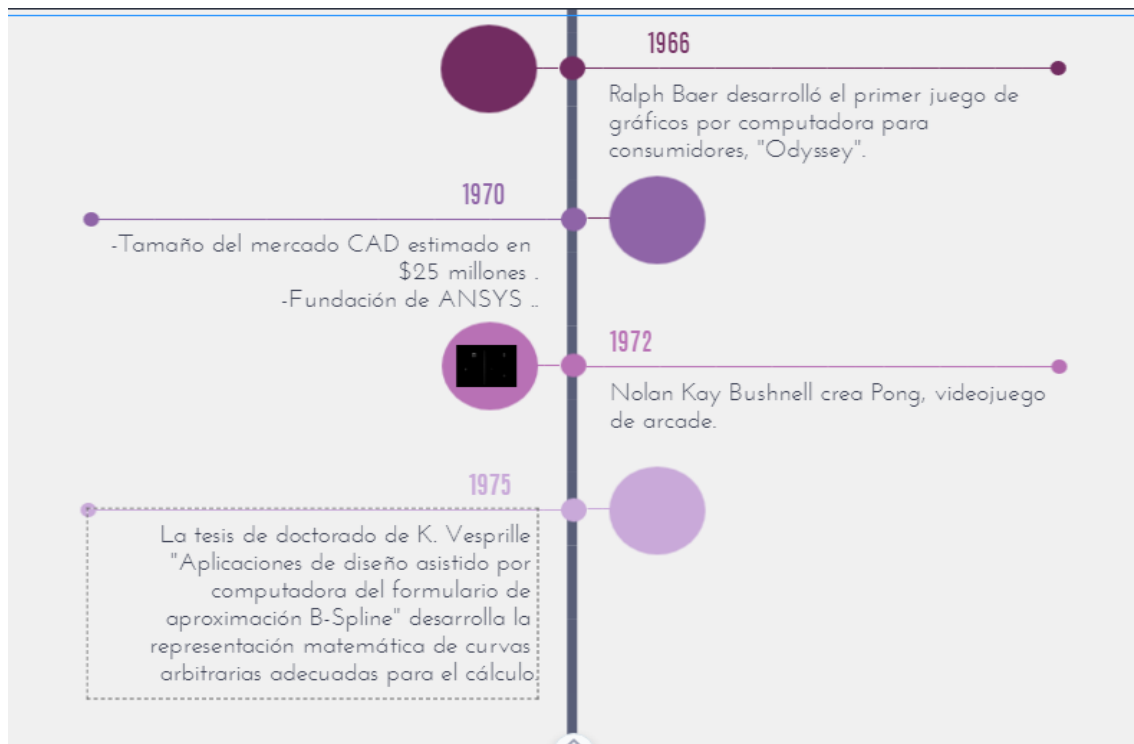
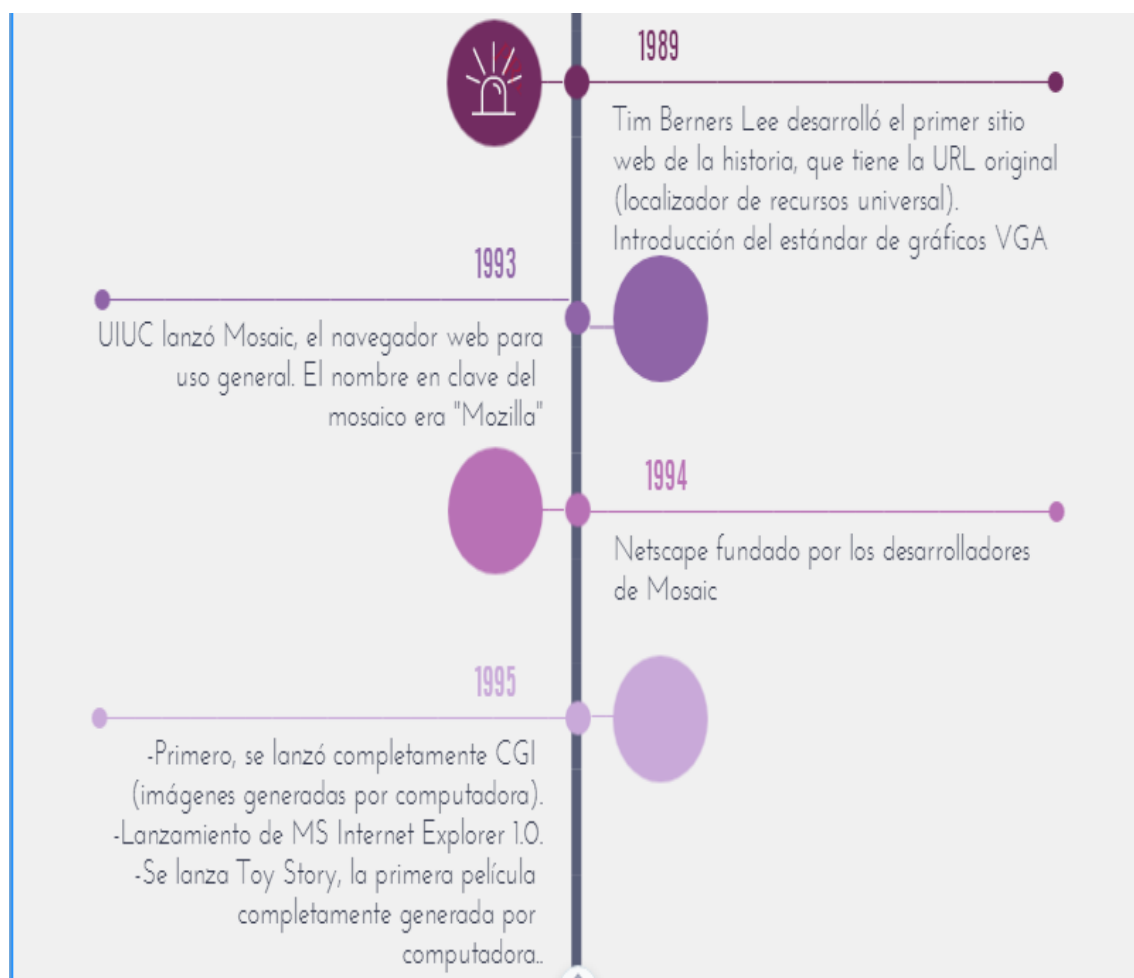
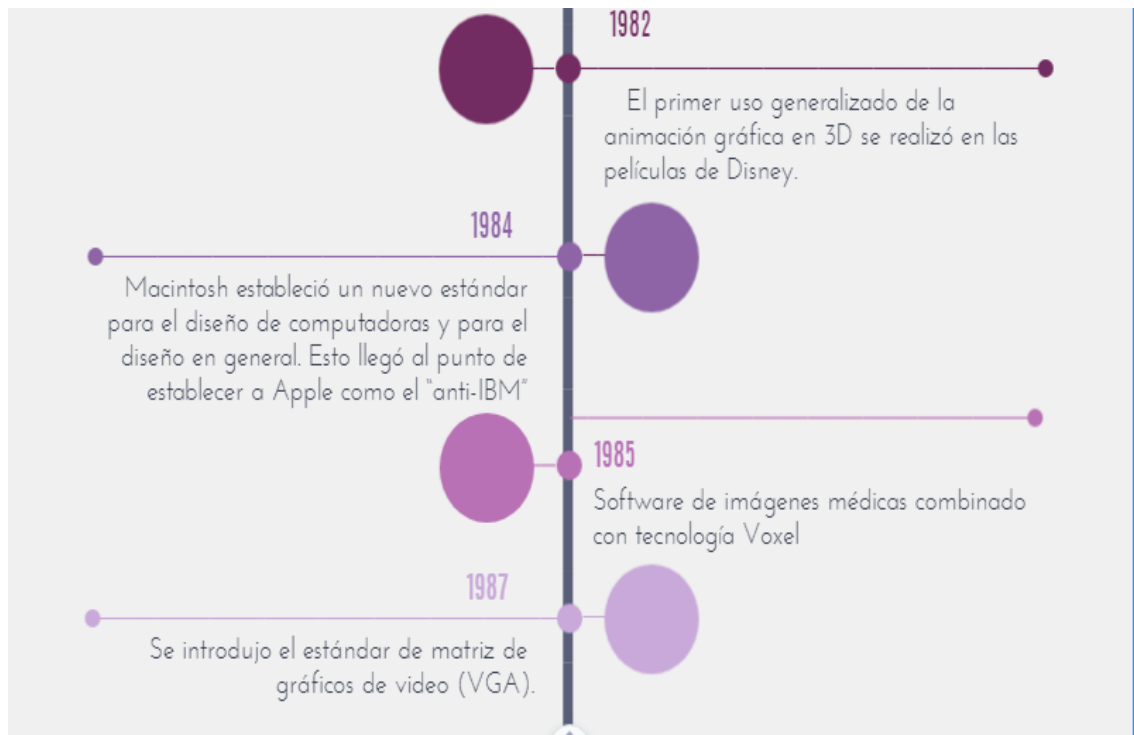
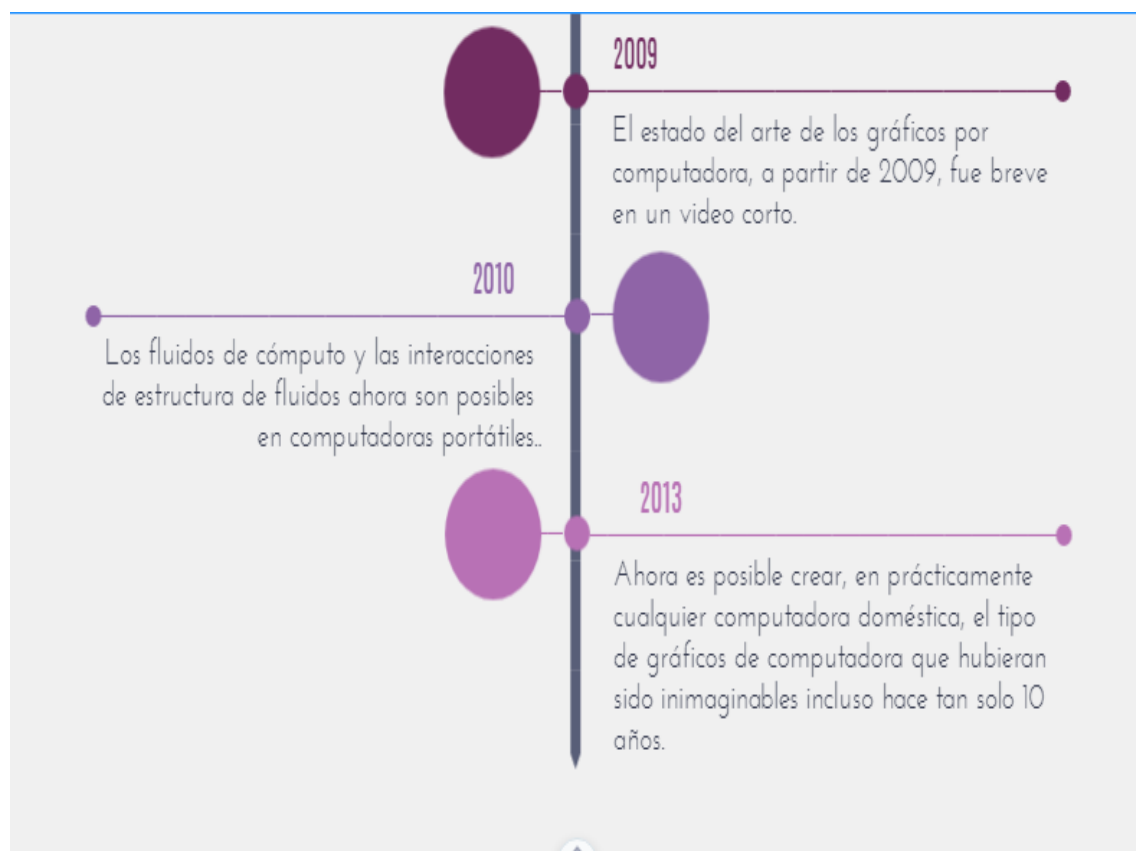
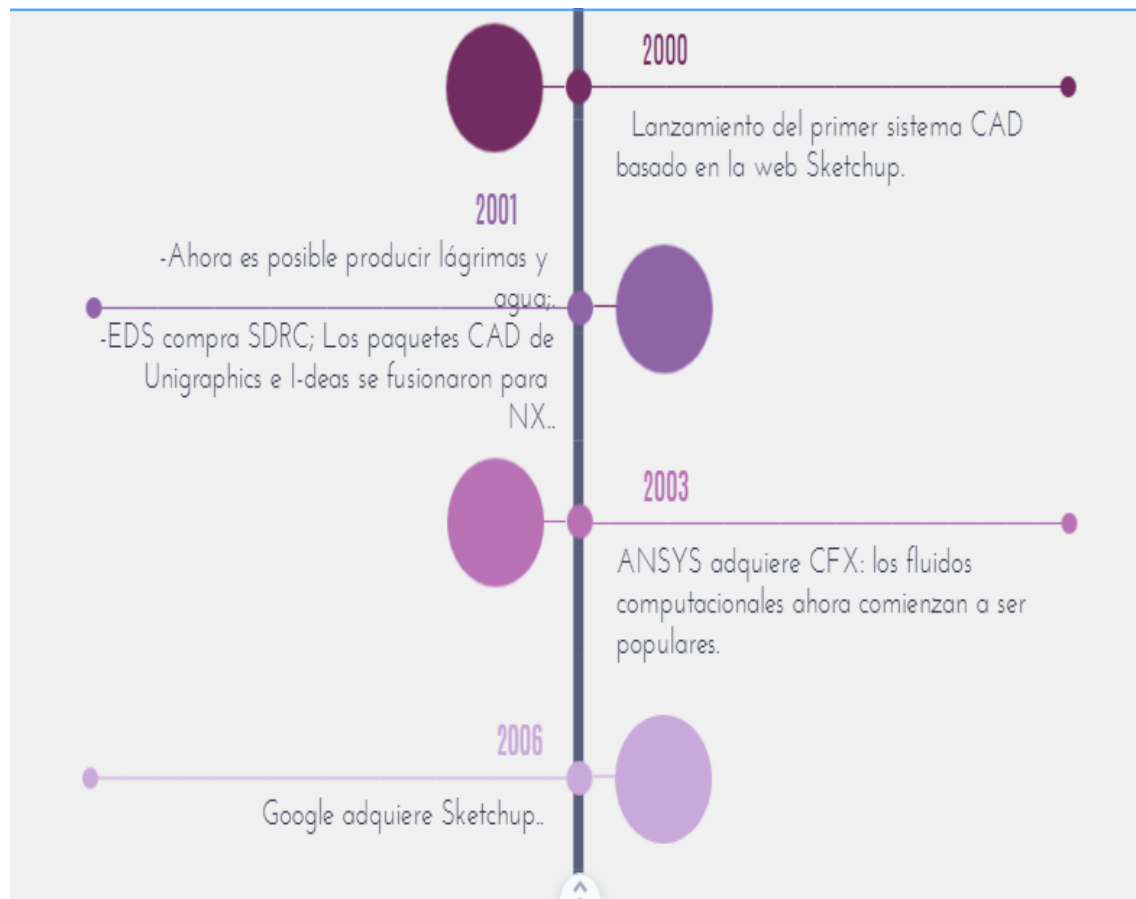


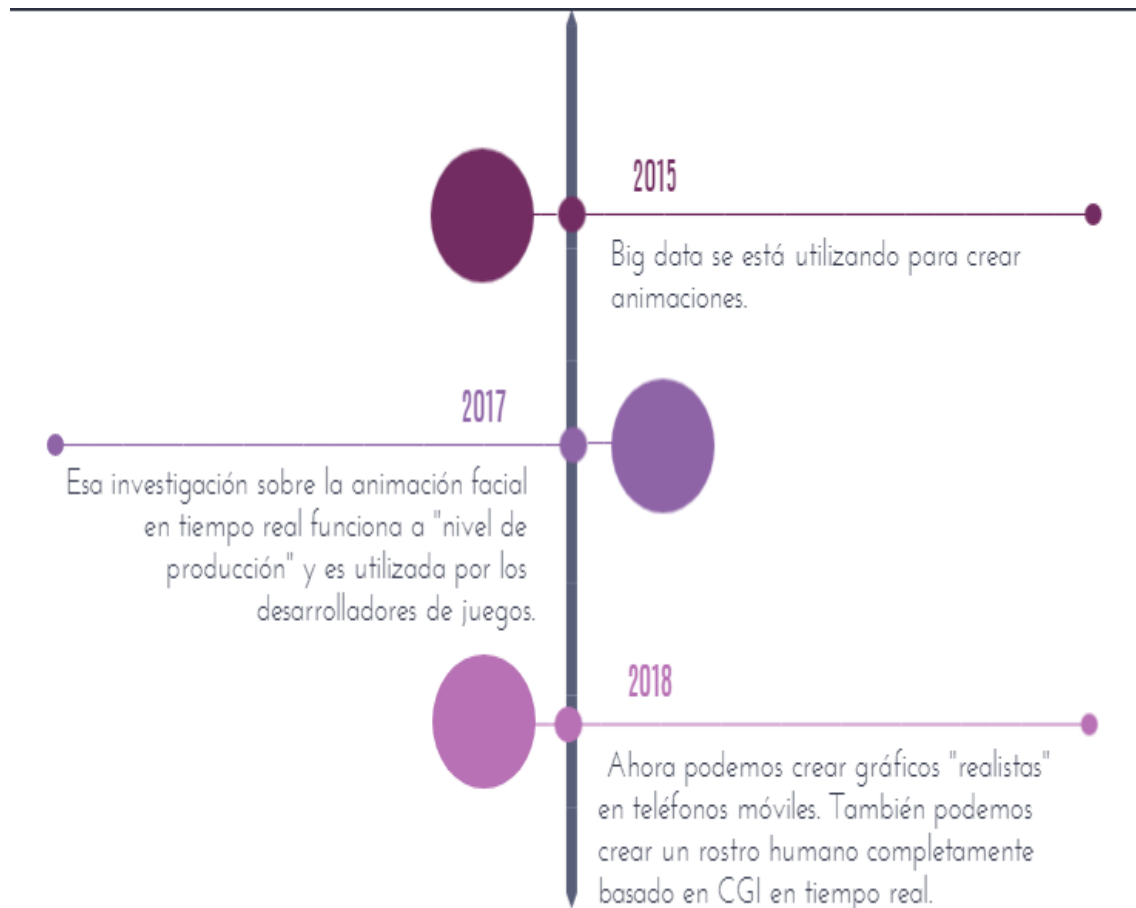
LINEA DEL TIEMPO COMPUTACION GRAFICA











Bibliografía:

1. https://deseng.ryerson.ca/dokuwiki/mec222:brief_history_of_computer_graphics
2. <https://www.tutorialandexample.com/history-of-computer-graphics/>
3. <https://www.cs.cmu.edu/~ph/nyit/masson/history.htm>
4. <http://acercadecomputaciongrafica.blogspot.com/2011/04/breve-historia-de-la-computacion.html>
5. <https://cursa.ihmc.us/rid=1LFHTWLZX-16PS4Z9-1K63/HISTORIA%20DE%20LA%20COMPUTACION%20GRAFICA.cmap>

Primer código

Lenguaje C:

```
1 #include <graphics.h>
2 #include <stdio.h>
3
4
5 int main()
6 {
7
8     int gd = DETECT, gm, color;
9
10    initgraph(&gd, &gm, "");
11
12
13    putpixel(20, 100, WHITE);
14
15
16    getch();
17
18
19    closegraph();
20
21    return 0;
22 }
```

La librería "graphics.h" permite graficar pixeles

Gd es Graphics Drivers, DETECT le pide al compilador que detecte los controladores gráficos.
le pide al compilador

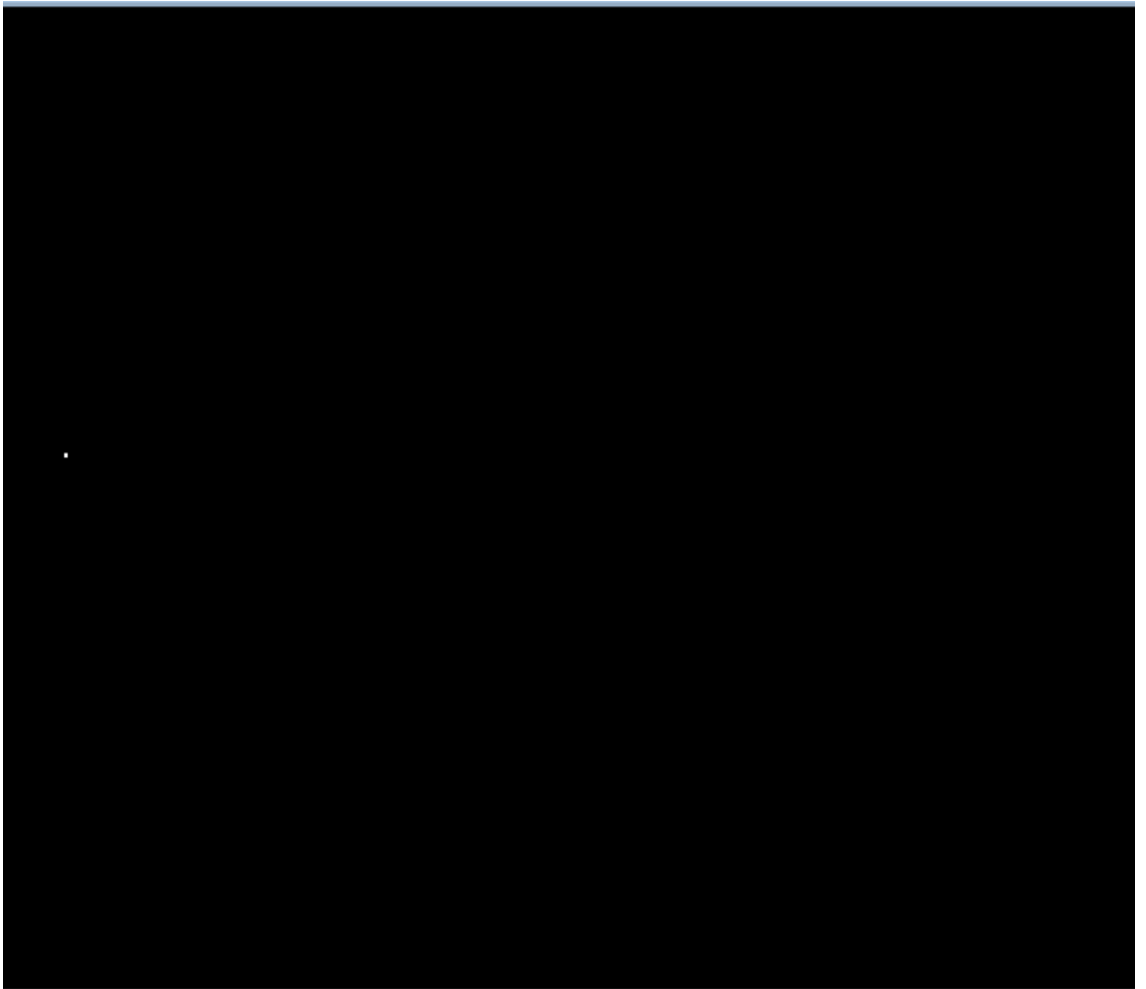
Sirve para escoger el color del pixel.

Gm es Graphics Mode, sirve para dar referencia a la resolución usada

La función initgraph sirve para iniciar los controladores usados.

La función closegraph cierra el modo gráfico.

La función putpixel es la función principal, se le asigna la posición X y Y ,y el color del pixel



Segundo código

Lenguaje Java:

```
1  package pintarpixel;
2
3  import java.awt.event.MouseAdapter;
4  import java.awt.event.MouseEvent;
5  import javax.swing.JFrame;
6  import javax.swing.JPanel;
7
8  public class PintarPixel {
9
10     public static void main(String[] args) {
11         JFrame f = new JFrame("Pintar Pixel");
12         f.setSize(600, 400); //tamaño de la ventana
13         f.setLocationRelativeTo(null); //centrado
14         f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
15         f.setVisible(true); //hacer visible la ventana
16
17         JPanel p = new JPanel();
18         p.setSize(600, 400); //tamaño del panel
19         f.add(p); //agrega a la ventana el panel creado
20
21         //Evento de click al panel
22         p.addMouseListener(new MouseAdapter() {
23             @Override
24             public void mouseClicked(MouseEvent e) {
25                 super.mouseClicked(e); //To change body
26                 //capturamos del evento e la posición x,y
27                 int x = e.getX();
28                 int y = e.getY();
29                 //pintamos el pixel en el panel
30                 p.getGraphics().drawLine(x, y, x, y);
31             }
32         });
33     }
34 }
```