

# PROYECTO DE PROGRAMACIÓN-TERCERA ENTREGA

## Presentado por:

Diego Cortés Mendoza – [dicortesm@unal.edu.co](mailto:dicortesm@unal.edu.co)

Germán Chacón Pedraza – [gchacon@unal.edu.co](mailto:gchacon@unal.edu.co)

Allison Castellanos Hernández - [alcastellanosh@unal.edu.co](mailto:alcastellanosh@unal.edu.co)

## Profesora:

Stephanie Torres Jiménez

[sttorresji@unal.edu.co](mailto:sttorresji@unal.edu.co)

Lunes 6 de Julio



**Universidad Nacional de Colombia**

**Facultad de Ingeniería**

**Departamento de Ingeniería Eléctrica y Electrónica**

**2020**

## **TABLA DE CONTENIDO**

<b>INTRODUCCION.....</b>	<b>3</b>
<b>CAMPO DE ACCION.....</b>	<b>3</b>
<b>DEFINICION GENERAL DEL PROYECTO.....</b>	<b>3</b>
<b>OBJETIVOS.....</b>	<b>4</b>
OBJETIVO GENERAL.....	4
OBJETIVO ESPECIFICO.....	4
<b>RESULTADO ESPERADO.....</b>	<b>4</b>
<b>RESULTADO OBTENIDO.....</b>	<b>4</b>
<b>DEMOSTRACION DEL PROYECTO.....</b>	<b>5</b>
<b>TRABAJO A FUTURO.....</b>	<b>19</b>
<b>CONCLUSIONES.....</b>	<b>19</b>
<b>BIBLIOGRAFIA.....</b>	<b>19</b>

## **INTRODUCCIÓN**

Teniendo cuenta la revolución informática de la última década, poco a poco los videojuegos se han ido inculcando en la cultura juvenil, así formando parte del día a día de muchos estudiantes tanto escolares como universitarios, inclusive según estudios presentados en la revista semana afirman que más de la mitad de los jóvenes juegan dichos videojuegos, es por ello que decidimos desarrollar uno que se vea directamente relacionado con la vida de estos últimos. Cabe resaltar la inspiración para su desarrollo, esta tomada del videojuego Lapse. Con el objetivo de llegar a este fin, haremos uso del programa Python para lograr evidenciar las habilidades adquiridas en lo corrido del semestre, entre esto se encuentran temáticas como: estructuras condicionales, bucles, operaciones aritméticas, matrices, archivos y diccionarios, además de algunas librerías para hacer mas llamativo el mismo

## **CAMPO DE ACCION**

Este proyecto será desarrollado como un videojuego, su campo de acción puede definirse como el ocio, entretenimiento y/o distracción.

## **DEFINICION GENERAL DEL PROYECTO**

El juego a programar tratará acerca de las posibles decisiones que podría tomar un estudiante de la Universidad Nacional en su día a día, y como estas serían capaces de influir en 4 ámbitos que consideramos mayoritariamente influyentes en la vida del mismo, tales como: Social, académico, emocional/salud y económico.

Al iniciar el juego, el usuario tendrá la posibilidad de personalizar sus estadísticas en los 4 ámbitos mencionados donde tendrá 200 puntos para distribuir entre los mismos manteniendo límites dentro de un rango lógico y realista el cual abarcaría  $1 < \text{rango} < 99$ . Si el usuario no lo decide así podrá usar estadísticas predeterminadas, de forma que las 4 mencionadas tendrían un valor de 50.

Temporalmente se irá progresando semanalmente con el fin de culminar exitosamente el semestre (16 semanas); el juego terminará antes lo previsto en caso de que alguno de los 4 ámbitos salga del rango  $1 > \text{rango} > 99$ .

## **OBJETIVOS**

### **OBJETIVO GENERAL**

Con el proyecto realizado, principalmente buscamos implementar de manera funcional y practica el conocimiento adquirido a lo largo del periodo académico, que, para hacerlo mas ameno se estructura en un juego de decisiones reales, buscando hacerlo lo mas amigable con el usuario y al tiempo traduce operaciones computacionales en entretenimiento.

### **OBJETIVO ESPECIFICO**

Como mencionamos anteriormente, este proyecto se ve enfocado hacia el ocio al tratarse un videojuego, pero debido a su estructuración y desarrollo del mismo, se encuentra de igual forma en la capacidad de evaluar la capacidad del usuario de tomar decisiones tanto reales como comunes en la Universidad Nacional, así mismo incluyendo temas como la organización del tiempo, relaciones sociales e inclusive el cuidado personal.

### **RESULTADO ESPERADO**

Después de tener la fase beta del videojuego y distribuirla entre los estudiantes, buscamos entretener a los mismos y permitirles visualizar por medio del mismo, como las pequeñas decisiones nos afectan en la vida cotidiana, que posterior a diversas mejoras se desea incluir una interfaz gráfica haciendo uso de librerías externas.

### **RESULTADO OBTENIDO**

A lo largo de este periodo de tiempo, se han desarrollado las diversas mejoras, así consiguiendo la previamente mencionada interfaz gráfica, donde el usuario podrá desarrollar el programa con detalles visuales que buscan dar ese toque de entretenimiento y una mayor calidad al implementar los conocimientos adicionales obtenidos.

## DEMOSTRACION DEL PROYECTO

```
def main():
    """
    Inicializa el diccionario "situaciones", la pantalla del modulo tkinter "screen_inicio", el diccionario
    "dicc_imagenes", el diccionario "dicc_estadisticas", las variables de tipo entero del modulo tkinter
    "Academico", "Salud", "Social", "Economico", y la variable de tipo cadena "txt_situacion".
    Invoca a la funcion menu().
    """
    global txt_situacion, situacion_a_usar
    situaciones = {0: textos_situacion_tutor, 1: textos_situacion_tutor2, 2: textos_situacion_tutor3,
                  3: situacion_jugar,
                  4: situacion_bicirrun, 5: situacion_amigos, 6: situacion_prestamo, 7: situacion_danza,
                  8: situacion_tropel,
                  9: situacion_comer, 10: situacion_parcial, 11: situacion_copia, 12: situacion_parcial_biblioteca,
                  13: situacion_freud,
                  50: situacion_tropel_1, 51: situacion_comer_1, 52: situacion_comer_2, 53: situacion_freud_1,
                  54: situacion_freud_2, 55: situacion_freud_3, 56: situacion_freud_4, 57: situacion_freud_5,
                  58: situacion_freud_6,
                  59: situacion_freud_7, 60: situacion_freud_8, 61: situacion_freud_9}
    situacion_a_usar = -1

    # ----- Ventana -----
    global screen_inicio, resultado
    screen_inicio = Tk()
    screen_inicio.resizable(False, False)
    screen_inicio.geometry("600x400+100+100")
    screen_inicio.title("Sobreviviendo en la UN")
    screen_inicio.config(bg="black")
    resultado = False

    dicc_imagenes = {}
    dicc_imagenes["imagen"] = PhotoImage(file="rsz_tutor.png")
    dicc_imagenes["coordenadas_imagen"] = (277, 210)
    dicc_imagenes["imagen_1"] = PhotoImage(file="rsz_tutor.png")
    dicc_imagenes["coordenadas_imagen_1"] = (277, 210)
    dicc_imagenes["imagen_2"] = PhotoImage(file="rsz_tutor.png")
    dicc_imagenes["coordenadas_imagen_2"] = (277, 210)

    Academico, Salud, Social, Economico, semana = IntVar(value=0), IntVar(value=0), IntVar(value=0), IntVar(value=0), IntVar(value=0)
    dicc_estadisticas = {"Academico": Academico, "Salud": Salud, "Social": Social, "Economico": Economico, "semana": semana}
    txt_situacion = StringVar()

    menu(screen_inicio, dicc_imagenes, dicc_estadisticas, situaciones)
    screen_inicio.mainloop()

def logo_bienvenida():
    canvvas = Canvas(master = screen_inicio, width = 600, height = 400)
    ventana = turtle.TurtleScreen(canvvas)
    ventana.bgcolor("black")
    canvvas.pack()
    Label(canvvas, text="CARGANDO", font=("Courier New", 15), bg="black", fg="DeepSkyBlue").place(x = 240, y = 310)
    Label(canvvas, text="Sobreviviendo en la UN", font=("Courier New", 20), bg="black", fg="yellow").place(x = 120, y = 20)
    grosor = 1
    hola = turtle.RawTurtle(ventana)
    hola.hideturtle()
    hola.speed(0)
    hola.pensize(grosor)
    head = hola.heading()
    cargando = turtle.RawTurtle(ventana)
    cargando.hideturtle()
    cargando.pensize(10)
    cargando.penup()
    cargando.goto(-240, -90)
    cargando.right(head)
    cargando.pendown()
    cargando.color("white")
    hola.penup()
    hola.goto(-120, 50)
    hola.right(head)
    hola.left(90)
    hola.pendown()
    z = 0
    pintar_S(hola, cargando, 1, z)
    grosor = 5
    hola.hideturtle()
    hola.penup()
    hola.goto(-40, 77)
    hola.right(head)
    hola.right(180)
```



Función main, inicializando estructuras y variables a utilizar, adicionalmente funciones encargadas del logo y barra de carga; en la función logo\_bienvenida se llaman las diversas letras.

```

def menu(screen, dicc_imagenes, dicc_estadisticas, situaciones):
    """
    Despliega textos para que el usuario escoja 3 posibles opciones e invoque a la funcion
    correspondiente, jugar en el modo estandar "tutorial_estandar()", jugar en el modo
    personalizado "tutorial_per()" o ver el puntaje maximo actual "v_puntajes()".

    parametros:
        screen:
        dicc_imagenes:
        dicc_estadisticas:
        situaciones:
    """
    trampa = False
    espacio = Frame(screen, width="500", height=400)
    espacio.config(bg="black")
    espacio.pack()

    Label(espacio, text="Sobreviviendo en la UN", bg="black", fg="gold", font=("Courier New", 18)).place(x="100",
                                                                                                     y="50")
    Label(espacio, text="Escoge tu modo de juego", bg="black", fg="gold", font=("Courier New", 12)).place(x="150",
                                                                                                     y="100")

    j_estandar = Button(espacio, text="Juego estandar", font=("Courier New", 10), fg="gold", bg="gray22",
                        command= lambda : tutorial_est(dicc_imagenes, dicc_estadisticas, situaciones, trampa))
    j_estandar.place(x=200, y=150)

    j_perso = Button(espacio, text="Juego personalizado", font=("Courier New", 10), fg="gold", bg="gray22",
                    command= lambda : tutorial_per(dicc_imagenes, dicc_estadisticas, situaciones, trampa))
    j_perso.place(x=180, y=190)

    puntajes = Button(espacio, text="Puntajes", font=("Courier New", 10), fg="gold", bg="gray22", command=v_puntajes)
    puntajes.place(x= 225, y =230)

    Label(espacio, text="V. 21.05.20", bg="black", fg="gold", font=("Courier New", 10)).place(x="50", y="290")
    Label(espacio, text="Created by:", bg="black", fg="gold", font=("Courier New", 10)).place(x="50", y="310")
    Label(espacio, text="Allison Castellanos Camilo Chacon Diego Cortes", bg="black", fg="gold",
          font=("Courier New", 10)).place(x="50", y="330")
    Label(espacio, text="Programacion de computadores 2020-1 UNAL", bg="black", fg="gold",
          font=("Courier New", 10)).place(x="50", y="350")

```



Función menú, encargada de dar las diversas opciones e información al usuario

```
def continuar(dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos):
    """
    Invoca a una situacion dependiendo de la semana en la cual se encuentre el usuario, o, si
    el usuario se encuentra actualmente en situaciones extensas.
    Aumenta en uno el valor de la llave "semana" del diccionario "dicc_ambitos".
    Comprueba si el usuario no ha perdido en la situacion que recién acaba de transcurrir del juego,
    de haber perdido, invoca a la funcion "game_over_total".

    parametros:
        dicc_tortugas: diccionario que contiene las variables del modulo turtle
        dicc_imagenes: diccionario que contiene las coordenadas y nombres de las imagenes de la
        situacion actual
        dicc_ambitos: diccionario que contiene las variables de cada ambito del usuario
        dicc_estadisticas: diccionario que contiene las variables del modulo tkinter las
        cuales se proyectan en pantalla
        situaciones: diccionario que contiene todas las posibles situaciones del juego
        dicc_textos: diccionario que contiene los mensajes que se proyectan en pantalla
    """
    global situacion_a_usar, situacion_extensa, resultado
    resultado = False
    if situacion_a_usar < 3:
        situacion_a_usar += 1
        situaciones[situacion_a_usar](dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos)
    elif dicc_ambitos["Social_0"] < 1 or dicc_ambitos["Social_0"] > 99 or dicc_ambitos["Academico_0"] < 1 or dicc_ambitos["Academico_0"] > 99 \
        or dicc_ambitos["Salud_0"] < 1 or dicc_ambitos["Salud_0"] > 99 or dicc_ambitos["Economico_0"] < 1 or dicc_ambitos["Economico_0"] > 99 or dicc_ambitos["semana_0"] >= 16:
        game_over_total(dicc_imagenes, dicc_ambitos, dicc_estadisticas, dicc_textos)
    elif situacion_extensa > 49:
        situaciones[situacion_extensa](dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos)
    elif dicc_ambitos["semana_0"] == 5 or dicc_ambitos["semana_0"] == 8 or dicc_ambitos["semana_0"] == 12:
        situacion_a_usar = random.randint(10, 12)
        dicc_ambitos["semana_0"] += 1
        dicc_estadisticas["semana"].set(dicc_ambitos["semana_0"])
        situaciones[situacion_a_usar](dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos)
    elif dicc_ambitos["semana_0"] == 6:
        situacion_a_usar = 13
        dicc_ambitos["semana_0"] += 1
        dicc_estadisticas["semana"].set(dicc_ambitos["semana_0"])
        situaciones[situacion_a_usar](dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos)
    else:
        situacion_a_usar = random.randint(3, 9)
        dicc_ambitos["semana_0"] += 1
        dicc_estadisticas["semana"].set(dicc_ambitos["semana_0"])
        situaciones[situacion_a_usar](dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos)
```



Función continuar, encargada de la variación de las situaciones según si estas son extensas, simples o de tutorial.



```

def v_situaciones_1(dicc_tortugas,dicc_imagenes,dicc_ambitos,dicc_estadisticas,situaciones,dicc_textos):
    """
    Despliega en pantalla los textos e imagenes de cada situacion, muestra
    graficamente los valores de las estadisticas, y habilita los botones opcion_1,
    opcion_2, opcion_3 dependiendo de si se quiere mostrar una situacion, o el
    resultado de una situacion.
    Si la situacion actual es la primera del programa, se inicializa el diccionario
    "dicc_tortugas" mediante la funcion "iniciar_tortugas()".
    El boton "opcion_1" y "opcion_2" invocan a la funcion "eleccion_1()" y
    "eleccion_2()", y el boton "opcion_3" invoca a la funcion "continuar()"

    parametros:
        dicc_tortugas: diccionario que contiene las variables del modulo turtle.
        dicc_imagenes: diccionario que contiene las coordenadas y nombres de las imagenes de la
        situacion actual.
        dicc_ambitos: diccionario que contiene las variables de cada ambito del usuario.
        dicc_estadisticas: diccionario que contiene las variables del modulo tkinter las
        cuales se proyectan en pantalla.
        situaciones: diccionario que contiene todas las posibles situaciones del juego.
        dicc_textos: diccionario que contiene los mensajes que se proyectan en pantalla .
    """
    global situacion_a_usar, screen_situaciones
    if situacion_a_usar == -1:
        screen_tutorial.withdraw()
        screen_situaciones = Toplevel()
        screen_situaciones.geometry("600x400+100+100")
        screen_situaciones.config(bg="black")
        puntaje_academico = dicc_tortugas["academico_inicial"]
        puntaje_economico = dicc_tortugas["economico_inicial"]
        puntaje_salud = dicc_tortugas["salud_inicial"]
        puntaje_social = dicc_tortugas["social_inicial"]
        dicc_tortugas = iniciar_tortugas()
        avanzar1(dicc_tortugas["AcademicoT"],puntaje_academico,dicc_tortugas["color_Academico"])
        avanzar1(dicc_tortugas["EconomicoT"],puntaje_economico,dicc_tortugas["color_Economico"])
        avanzar1(dicc_tortugas["SocialT"],puntaje_social,dicc_tortugas["color_Social"])
        avanzar1(dicc_tortugas["SaludT"],puntaje_salud,dicc_tortugas["color_Salud"])

    space_est = Frame(screen_situaciones, width=85, height=80)
    space_est.place(x = 0 , y =0)
    space_est.config(bg="black")

    space_est2 = Frame(screen_situaciones, width=50, height=100)
    space_est2.place(x = 550 , y =0)
    space_est2.config(bg="black")

    Label(space_est, text="academico", font=("Courier New", 9), bg="black", fg="DeepSkyBlue").place(x =0, y =0)
    Label(space_est, text="salud", font=("Courier New", 9), bg="black", fg="gold").place(x =0, y =15)
    Label(space_est, text="social", font=("Courier New", 9), bg="black", fg="green").place(x =0, y =30)
    Label(space_est, text="economico", font=("Courier New", 9), bg="black", fg="red").place(x =0, y =45)
    Label(space_est, text="semanas ", font=("Courier New", 9), bg="black", fg="silver").place(x =0, y =60)

    Label(space_est2, textvariable=dicc_estadisticas["Academico"], font=("Courier New", 9), bg="black", fg="DeepSkyBlue").place(x =10, y =0)
    Label(space_est2, textvariable=dicc_estadisticas["Salud"], font=("Courier New", 9), bg="black", fg="gold").place(x =10, y =15)
    Label(space_est2, textvariable=dicc_estadisticas["Social"], font=("Courier New", 9), bg="black", fg="green").place(x =10, y =30)
    Label(space_est2, textvariable=dicc_estadisticas["Economico"], font=("Courier New", 9), bg="black", fg="red").place(x =10, y =45)
    Label(space_est2, textvariable=dicc_estadisticas["semana"], font=("Courier New", 10), bg="black", fg="silver").place(x =10, y =60)

    #500
    space_situation = Frame(screen_situaciones, width=480, height=300)
    space_situation.place(x = 80 , y = 60)
    space_situation.config(bg="black")
    # Imagen situaciones

    Label(screen_situaciones, image=dicc_imagenes["imagen"]).place(x=dicc_imagenes["coordenadas_imagen"][0], y= dicc_imagenes["coordenadas_imagen"][1])

    # Texto situaciones
    Label(space_situation, textvariable=dicc_textos["txt_situacion"], font=("Courier New", 11), bg="gold", fg="black",
        justify=LEFT).place(x=10, y=10)

```

```

if situacion_a_usar < 3:
    opcion_1 = Button(space_situation, text="(1)", font=("Courier New", 10), fg="gold", bg="gray22",
                      command= lambda : eleccion_1(dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos), width=10)
    opcion_1.place(x=80, y=270)

    opcion_2 = Button(space_situation, text="(2)", font=("Courier New", 10), fg="gold", bg="gray22",
                      command= lambda : eleccion_2(dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos), width=10)
    opcion_2.place(x=280, y=270)

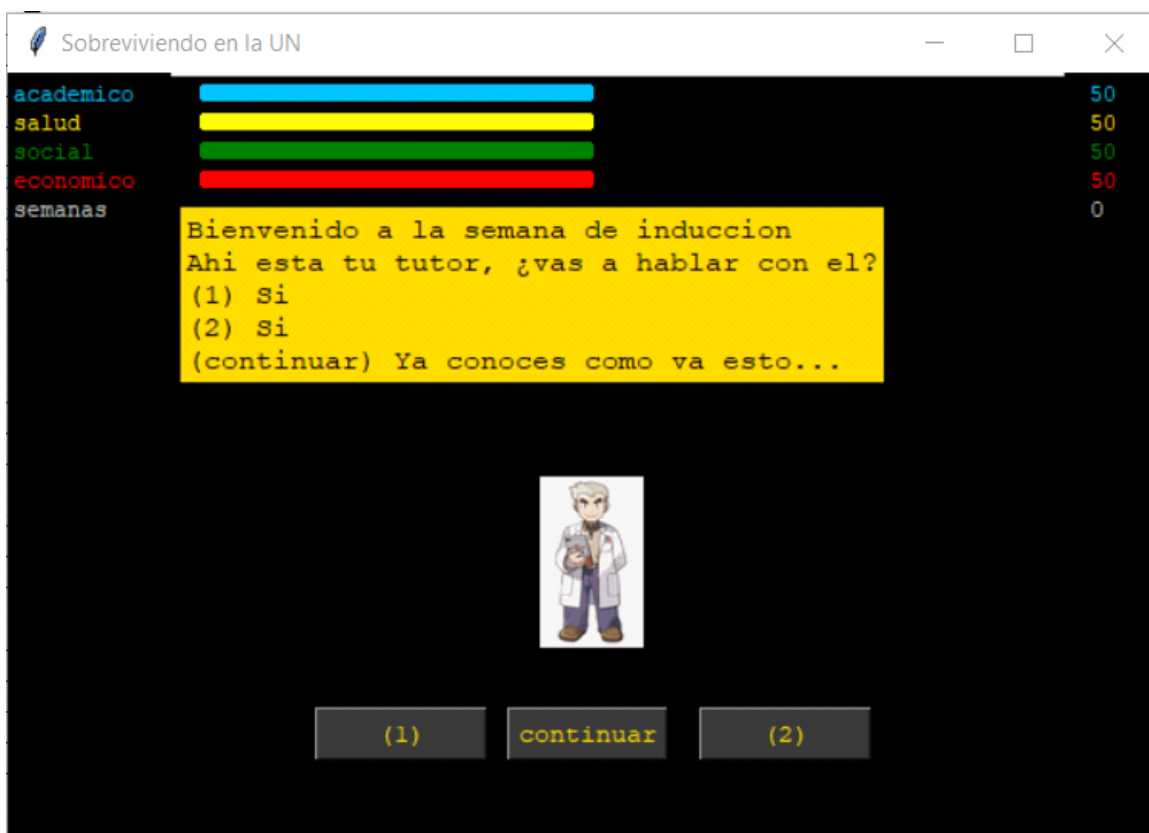
    opcion_3 = Button(space_situation, text="continuar", font=("Courier New", 10), fg="gold", bg="gray22",
                      command= lambda : situacion_jugar(dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos))
    opcion_3.place(x= 180, y = 270)

else:
    if resultado == False:
        opcion_1 = Button(space_situation, text="(1)", font=("Courier New", 10), fg="gold", bg="gray22",
                          command= lambda : eleccion_1(dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos), width=10)
        opcion_1.place(x=80, y=270)

        opcion_2 = Button(space_situation, text="(2)", font=("Courier New", 10), fg="gold", bg="gray22",
                          command= lambda : eleccion_2(dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos), width=10)
        opcion_2.place(x=280, y=270)

    else:
        opcion_3 = Button(space_situation, text="continuar", font=("Courier New", 10), fg="gold", bg="gray22",
                          command= lambda : continuar(dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos))
        opcion_3.place(x= 180, y = 270)

```



Función `v_situaciones_1`, encargada de proyectar la información requerida por cada situación, además de habilitar los botones.

```

def iniciar_tortugas():
    """
    Inicializa todas las variables del modulo turtle y las guarda en un diccionario

    Inicializa las tortugas que representaran los valores de las estadísticas del usuario,
    junto con sus características, tales como su color, su grosor, su velocidad y su
    posición.

    retorno:
    """
    dicc_tortugas:
    """
    dicc_tortugas = {}
    canvass = Canvas(master = screen_situaciones, width = 470 , height = 60)
    pantalla_turtle = turtle.TurtleScreen(canvass)
    pantalla_turtle.bgcolor("black")
    canvass.place( x = 80 , y = 0)

    velocidad = 10000000
    grosor = 5

    dicc_tortugas["color_Academico"] = "DeepSkyBlue"
    dicc_tortugas["AcademicoT"] = turtle.RawTurtle(pantalla_turtle)
    dicc_tortugas["AcademicoT"].hideturtle()
    dicc_tortugas["AcademicoT"].pensize(grosor)
    dicc_tortugas["AcademicoT"].speed(velocidad)
    head = dicc_tortugas["AcademicoT"].heading()

    dicc_tortugas["color_Economico"] = "red"
    dicc_tortugas["EconomicoT"] = turtle.RawTurtle(pantalla_turtle)
    dicc_tortugas["EconomicoT"].hideturtle()
    dicc_tortugas["EconomicoT"].pensize(grosor)
    dicc_tortugas["EconomicoT"].speed(velocidad)

    dicc_tortugas["color_Salud"] = "yellow"
    dicc_tortugas["SaludT"] = turtle.RawTurtle(pantalla_turtle)
    dicc_tortugas["SaludT"].hideturtle()
    dicc_tortugas["SaludT"].pensize(grosor)
    dicc_tortugas["SaludT"].speed(velocidad)

    dicc_tortugas["color_Social"] = "green"
    dicc_tortugas["SocialT"] = turtle.RawTurtle(pantalla_turtle)
    dicc_tortugas["SocialT"].hideturtle()
    dicc_tortugas["SocialT"].pensize(grosor)
    dicc_tortugas["SocialT"].speed(velocidad)

    dicc_tortugas["AcademicoT"].up()
    dicc_tortugas["AcademicoT"].goto(-215,24)
    dicc_tortugas["AcademicoT"].right(head)
    dicc_tortugas["EconomicoT"].up()
    dicc_tortugas["EconomicoT"].goto(-215,-21)
    dicc_tortugas["EconomicoT"].right(head)
    dicc_tortugas["SocialT"].up()
    dicc_tortugas["SocialT"].goto(-215,-6)
    dicc_tortugas["SocialT"].right(head)
    dicc_tortugas["SaludT"].up()
    dicc_tortugas["SaludT"].goto(-215,9)
    dicc_tortugas["SaludT"].right(head)

    dicc_tortugas["AcademicoT"].down()
    dicc_tortugas["EconomicoT"].down()
    dicc_tortugas["SaludT"].down()
    dicc_tortugas["SocialT"].down()
    return dicc_tortugas

```



Función iniciar\_tortugas, encargada de generar la ventana respectiva para la librería turtle.

```

def eleccion_1(dicc_tortugas,dicc_imagenes,dicc_ambitos,dicc_estadisticas,situaciones,dicc_textos):
    """
    Altera las estadísticas del usuario conforme a la decisión número uno

    Inicializa el Frame space_situacion para eliminar los textos, imágenes y botones de la
    situación anterior.
    Asigna nuevos valores a las estadísticas del usuario si es que la elección que tomo
    generaba cambios en ellas, de ser así, las tortugas del programa se desplazarán para
    ilustrar los nuevos valores de las estadísticas.
    Si el usuario no se encuentra en el tutorial, la función v_situaciones_1 es invocada.

    parametros:
        dicc_tortugas: diccionario que contiene las variables del modulo turtle
        dicc_imagenes: diccionario que contiene las coordenadas y nombres de las imágenes de la
        situación actual
        dicc_ambitos: diccionario que contiene las variables de cada ambito del usuario
        dicc_estadisticas: diccionario que contiene las variables del modulo tkinter las
        cuales se proyectan en pantalla
        situaciones: diccionario que contiene todas las posibles situaciones del juego
        dicc_textos: diccionario que contiene los mensajes que se proyectan en pantalla
    """
    global situacion_a_usar, situacion_extensa, situacion_extensa_1, resultado, screen_situaciones

    space_situacion = Frame(screen_situaciones, width=480, height=300)
    space_situacion.place(x = 80 , y = 60)
    space_situacion.config(bg="black")

    dicc_imagenes["imagen"] = dicc_imagenes["imagen_1"]
    dicc_imagenes["coordenadas_imagen"] = dicc_imagenes["coordenadas_imagen_1"]

    situacion_extensa = situacion_extensa_1

    dicc_estadisticas["Academico"].set(dicc_ambitos["Academico_1"])
    dicc_ambitos["Academico_0"] = dicc_ambitos["Academico_1"]
    dicc_estadisticas["Salud"].set(dicc_ambitos["Salud_1"])
    dicc_ambitos["Salud_0"] = dicc_ambitos["Salud_1"]
    dicc_estadisticas["Economico"].set(dicc_ambitos["Economico_1"])
    dicc_ambitos["Economico_0"] = dicc_ambitos["Economico_1"]
    dicc_estadisticas["Social"].set(dicc_ambitos["Social_1"])
    dicc_ambitos["Social_0"] = dicc_ambitos["Social_1"]

    #codigo de tortuga
    if situacion_a_usar >= 3:
        if dicc_tortugas["tortuga_academico1"] > 0:
            avanzar(dicc_tortugas["AcademicoT"],dicc_tortugas["tortuga_academico1"],dicc_tortugas["color_Academico"])
        elif dicc_tortugas["tortuga_academico1"] < 0:
            retroceder(dicc_tortugas["AcademicoT"],abs(dicc_tortugas["tortuga_academico1"]))

        if dicc_tortugas["tortuga_economico1"] > 0:
            avanzar(dicc_tortugas["EconomicoT"],dicc_tortugas["tortuga_economico1"],dicc_tortugas["color_Economico"])
        elif dicc_tortugas["tortuga_economico1"] < 0:
            retroceder(dicc_tortugas["EconomicoT"],abs(dicc_tortugas["tortuga_economico1"]))

        if dicc_tortugas["tortuga_salud1"] > 0:
            avanzar(dicc_tortugas["SaludT"],dicc_tortugas["tortuga_salud1"],dicc_tortugas["color_Salud"])
        elif dicc_tortugas["tortuga_salud1"] < 0:
            retroceder(dicc_tortugas["SaludT"],abs(dicc_tortugas["tortuga_salud1"]))

        if dicc_tortugas["tortuga_social1"] > 0:
            avanzar(dicc_tortugas["SocialT"],dicc_tortugas["tortuga_social1"],dicc_tortugas["color_Social"])
        elif dicc_tortugas["tortuga_social1"] < 0:
            retroceder(dicc_tortugas["SocialT"],abs(dicc_tortugas["tortuga_social1"]))

    #codigo de tortuga
    if situacion_a_usar < 3:
        continuar(dicc_tortugas,dicc_imagenes,dicc_ambitos,dicc_estadisticas,situaciones,dicc_textos)
    else:
        dicc_textos["txt_situacion"].set(dicc_textos["txt_1"])
        resultado = True
        v_situaciones_1(dicc_tortugas,dicc_imagenes,dicc_ambitos,dicc_estadisticas,situaciones,dicc_textos)

```



Función eleccion\_1, Encargada de variar las estadísticas tanto numéricamente como visualmente en turtle según la opción escogida(se llama a eleccion\_1 o eleccion\_2)

```

def avanzar(tortuga,valor,color_tortuga):
    """
    La tortuga avanzara serpenteando varias veces dependiendo del valor del parametro
    "valor" para aumentar el area de la barra que representa el valor de la
    estadisticas del usuario.

    parametros:
        tortuga:
        valor:
        color_tortuga:
    """
    tortuga.color(color_tortuga)
    tortuga.right(180)
    tortuga.forward(4)
    tortuga.right(90)
    for a in range(valor):
        for b in range(4):
            tortuga.right(90)
            tortuga.forward(4)
            tortuga.left(90)
            tortuga.forward(2.5)
            tortuga.left(90)
            tortuga.forward(4)
            tortuga.right(90)
            tortuga.forward(2.5)
    tortuga.right(90)
    tortuga.forward(4)

def retroceder(tortuga,valor):
    """
    La tortuga retrocedera serpenteando varias veces dependiendo del valor del parametro
    "valor" para disminuir el area de la barra que representa el valor de las estadisticas
    del usuario.

    parametros:
        tortuga:
        valor:
    """
    tortuga.right(90)
    tortuga.color("black")
    for c in range(valor):
        for d in range(4):
            tortuga.right(90)
            tortuga.forward(4)
            tortuga.left(90)
            tortuga.forward(2.5)
            tortuga.left(90)
            tortuga.forward(4)
            tortuga.right(90)
            tortuga.forward(2.5)
    tortuga.right(90)
    tortuga.forward(4)
    tortuga.right(180)
    tortuga.forward(4)

```



Función avanzar y retroceder, encargada del movimiento de la tortuga (librería turtle)

```

def situacion_tropel(dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos):
    global situacion_extensa_1

    dicc_imagenes["imagen"] = PhotoImage(file="tropel.png")
    dicc_imagenes["imagen_1"] = PhotoImage(file="tropel_1.png")
    dicc_imagenes["imagen_2"] = PhotoImage(file="tropel_2.png")
    dicc_imagenes["coordenadas_imagen"] = (205, 150)
    dicc_imagenes["coordenadas_imagen_1"] = (175, 110)
    dicc_imagenes["coordenadas_imagen_2"] = (160, 110)

    dicc_textos["txt_situacion"].set(
        "Vas de salida de la U pero hay un tropel\njusto por donde casi siempre sales\n(1)Quedarte a ver (?)\"
        "\n(2)Buscar otra salida (Nada de nada...)\"")
    dicc_textos["txt_1"] = "Vere solo un rato"
    dicc_textos["txt_2"] = "Das media vuelta y te vas a tu casa"
    situacion_extensa_1 = 50

    dicc_ambitos["Academico_1"] = dicc_ambitos["Academico_0"]
    dicc_ambitos["Academico_2"] = dicc_ambitos["Academico_0"]
    dicc_ambitos["Social_1"] = dicc_ambitos["Social_0"]
    dicc_ambitos["Social_2"] = dicc_ambitos["Social_0"]
    dicc_ambitos["Salud_1"] = dicc_ambitos["Salud_0"]
    dicc_ambitos["Salud_2"] = dicc_ambitos["Salud_0"]
    dicc_ambitos["Economico_1"] = dicc_ambitos["Economico_0"]
    dicc_ambitos["Economico_2"] = dicc_ambitos["Economico_0"]

    #codigo turtle
    dicc_tortugas["tortuga_academico1"] = 0
    dicc_tortugas["tortuga_academico2"] = 0
    dicc_tortugas["tortuga_social1"] = 0
    dicc_tortugas["tortuga_social2"] = 0
    dicc_tortugas["tortuga_salud1"] = 0
    dicc_tortugas["tortuga_salud2"] = 0

def situacion_tropel_1(dicc_tortugas, dicc_imagenes, dicc_ambitos, dicc_estadisticas, situaciones, dicc_textos):
    global situacion_extensa_1

    dicc_imagenes["imagen"] = PhotoImage(file="tropel_1_1.png")
    dicc_imagenes["imagen_1"] = PhotoImage(file="tropel_1_1_1.png")
    dicc_imagenes["imagen_2"] = PhotoImage(file="tropel_1_1_2.png")
    dicc_imagenes["coordenadas_imagen"] = (220, 150)
    dicc_imagenes["coordenadas_imagen_1"] = (190, 130)
    dicc_imagenes["coordenadas_imagen_2"] = (200, 130)

    dicc_textos["txt_situacion"].set(
        "Vere solo un rato\n(1)Ver de cerca (?)\"
        "\n(2)Ver de lejos (?)\"")
    dicc_textos["txt_1"] = "No paso nada..."
    dicc_textos["txt_2"] = "Te cayo una granada lacrimogena a tu lado..."
    situacion_extensa_1 = 0

    dicc_ambitos["Academico_1"] = dicc_ambitos["Academico_0"]
    dicc_ambitos["Academico_2"] = dicc_ambitos["Academico_0"]
    dicc_ambitos["Social_1"] = dicc_ambitos["Social_0"]
    dicc_ambitos["Social_2"] = dicc_ambitos["Social_0"]
    dicc_ambitos["Salud_1"] = dicc_ambitos["Salud_0"]
    dicc_ambitos["Salud_2"] = dicc_ambitos["Salud_0"] - 15
    dicc_ambitos["Economico_1"] = dicc_ambitos["Economico_0"]
    dicc_ambitos["Economico_2"] = dicc_ambitos["Economico_0"]

```

Existen situaciones simples y complejas; las simples solo constan de 1 posibilidad, en cambio, las extensas varían según decisiones del usuario.

```

def game_over_total(dicc_imagenes , dicc_ambitos,dicc_estadisticas,dicc_textos):
    if dicc_ambitos["Academico_0"] < 1:
        dicc_textos["txt_situacion"].set("Perdiste la calidad de estudiante, \ntu papa quedo por debajo de 3.0")
        dicc_imagenes["imagen"] = PhotoImage(file="game_over_academico_negativo.png")
        dicc_imagenes["coordenadas_imagen"] = (215, 140)
    elif dicc_ambitos["Academico_0"] > 99:
        dicc_textos["txt_situacion"].set("Te convertiste en un raton de biblioteca")
        dicc_imagenes["imagen"] = PhotoImage(file="game_over_academico_positivo.png")
        dicc_imagenes["coordenadas_imagen"] = (215, 140)
    elif dicc_ambitos["Salud_0"] < 1:
        dicc_textos["txt_situacion"].set("Te enfermas, eres hospitalizado y \nnno regresas. F")
        dicc_imagenes["imagen"] = PhotoImage(file="game_over_salud_negativo.png")
        dicc_imagenes["coordenadas_imagen"] = (215, 140)
    elif dicc_ambitos["Salud_0"] > 99:
        dicc_textos["txt_situacion"].set("Estas sano pero dejaste de \nlado lo demás")
        dicc_imagenes["imagen"] = PhotoImage(file="game_over_academico_positivo.png")
        dicc_imagenes["coordenadas_imagen"] = (215, 140)
    elif dicc_ambitos["Social_0"] < 1:
        dicc_textos["txt_situacion"].set("Te quedaste solo")
        dicc_imagenes["imagen"] = PhotoImage(file="game_over_social_negativo.png")
        dicc_imagenes["coordenadas_imagen"] = (215, 140)
    elif dicc_ambitos["Social_0"] > 99:
        dicc_textos["txt_situacion"].set("Te volviste muy social y dejaste \nlo demas abandonado")
        dicc_imagenes["imagen"] = PhotoImage(file="game_over_social_positivo.png")
        dicc_imagenes["coordenadas_imagen"] = (215, 140)
    elif dicc_ambitos["Economico_0"] < 1:
        dicc_textos["txt_situacion"].set("Te quedaste sin dinero para ir a \nla U")
        dicc_imagenes["imagen"] = PhotoImage(file="game_over_economico_negativo.png")
        dicc_imagenes["coordenadas_imagen"] = (215, 140)
    elif dicc_ambitos["Economico_0"] > 99:
        dicc_textos["txt_situacion"].set("Ahorraste sin tener en cuenta tu \nvida")
        dicc_imagenes["imagen"] = PhotoImage(file="game_over_economico_positivo.png")
        dicc_imagenes["coordenadas_imagen"] = (215, 140)

def pantalla_game_over(dicc_imagenes,dicc_estadisticas,dicc_textos):
    global screen_situaciones
    screen_situaciones.withdraw()
    screen_situaciones = Toplevel()
    screen_situaciones.geometry("600x400+100+100")
    screen_situaciones.config(bg="black")

    space_est = Frame(screen_situaciones, width=500, height=80)
    space_est.pack(padx=5)
    space_est.config(bg="black")

    Label(space_est, text="Academico ", font=("Courier New", 9), bg="black", fg="gold").grid(row=0, column=0)
    Label(space_est, text="Salud ", font=("Courier New", 9), bg="black", fg="gold").grid(row=0, column=1)
    Label(space_est, text="Social ", font=("Courier New", 9), bg="black", fg="gold").grid(row=0, column=2)
    Label(space_est, text="Economico ", font=("Courier New", 9), bg="black", fg="gold").grid(row=0, column=3)
    Label(space_est, text="Semanas ", font=("Courier New", 9), bg="black", fg="gold").grid(row=0, column=4)

    Label(space_est, textvariable=dicc_estadisticas["Academico"], font=("Courier New", 9), bg="black", fg="gold").grid(row=1, column=0)
    Label(space_est, textvariable=dicc_estadisticas["Salud"], font=("Courier New", 9), bg="black", fg="gold").grid(row=1, column=1)
    Label(space_est, textvariable=dicc_estadisticas["Social"], font=("Courier New", 9), bg="black", fg="gold").grid(row=1, column=2)
    Label(space_est, textvariable=dicc_estadisticas["Economico"], font=("Courier New", 9), bg="black", fg="gold").grid(row=1, column=3)
    Label(space_est, textvariable=dicc_estadisticas["semana"], font=("Courier New", 9), bg="black", fg="gold").grid(row=1, column=4)

    space_situacion = Frame(screen_situaciones, width=500, height=300)
    space_situacion.pack(padx=5, pady=10)
    space_situacion.config(bg="black")
    # imagenes resultantes
    Label(screen_situaciones, image=dicc_imagenes["imagen"]).place(x=dicc_imagenes["coordenadas_imagen"][0], y=dicc_imagenes["coordenadas_imagen"][1])
    # Texto situaciones
    Label(space_situacion, textvariable=dicc_textos["txt_situacion"], font=("Courier New", 12), bg="gold", fg="black",

```



```

Label(space_situacion, textvariable=Carrera, font=("Courier New", 12), bg="gold", fg="black",
      justify=LEFT).place(x=10, y=200)

record = open("records_usuario", "a+")
encontrado = False
perfiles, lineas_archivo = leer_puntajes()
for i in range(len(lineas_archivo)):
    if name == lineas_archivo[i][0]:
        encontrado = True
        puntaje_actual = (dicc_estadisticas["Academico"].get() + dicc_estadisticas["Salud"].get() + dicc_estadisticas["Social"] + dicc_economico) * dicc_estadisticas["Semana"]
        datos = [int(num) for num in lineas_archivo[i][2:6]]
        puntaje = sum(datos[:4]) * datos[4]
        if puntaje_actual > puntaje:
            lineas_archivo[i][2] = dicc_estadisticas["Academico"].get()
            lineas_archivo[i][3] = dicc_estadisticas["Salud"].get()
            lineas_archivo[i][4] = dicc_estadisticas["Social"].get()
            lineas_archivo[i][5] = dicc_estadisticas["Economico"].get()
            lineas_archivo[i][6] = dicc_estadisticas["Semana"].get()
            print(datos)
if encontrado == False:
    record.write("{}({},{},{},{},{})\n".format(name.get(), password(), dicc_estadisticas["Academico"].get(), dicc_estadisticas["Salud"].get(), dicc_estadisticas["Social"].get(), dicc_estadisticas["Economico"].get(), dicc_estadisticas["Semana"].get()))
else:
    record.seek(0)
    for lineas in lineas_archivo:
        record.write(lineas)
record.close()

global opcion_3
opcion_3 = Button(screen_situaciones, text="volver al menu", font=("Courier New", 10), fg="gold", bg="gray22",
                  command=lambda: [screen_inicio.destroy(), main()])
opcion_3.pack(side=BOTTOM)

```



Las funciones pantalla\_game\_over y game\_over\_total van en conjunto, esto se debe a que game\_over\_total verifica la razón de perdida enviando parámetros a pantalla\_game\_over y así su debida proyección.



```
def v_puntajes():
    perfiles, lista_archivos = leer_puntajes()
    screen_puntajes = Tk()
    screen_puntajes.resizable(False, False)
    screen_puntajes.geometry("600x400+100+100")
    screen_puntajes.title("Sobreviviendo en la UN")
    screen_puntajes.config(bg="black")

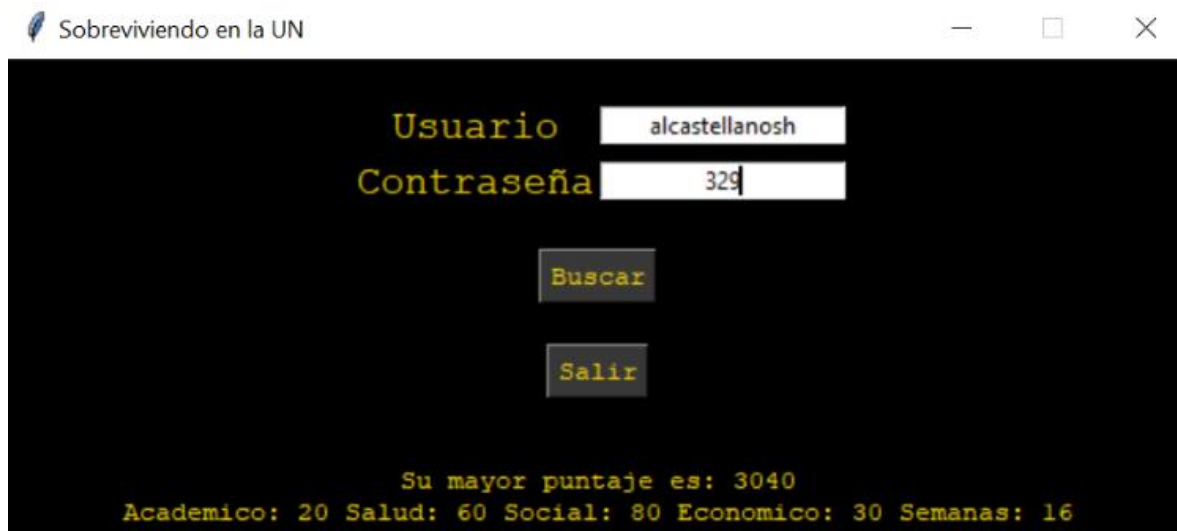
    datos = Frame(screen_puntajes, bg="black")
    datos.pack(pady=20)
    Label(datos, text="Usuario", bg="black", fg="gold", font=("Courier New", 15)).grid(row=0, column=0)
    Label(datos, text="Contraseña", bg="black", fg="gold", font=("Courier New", 15)).grid(row=1, column=0)

    entrada_usuario = StringVar(screen_puntajes)
    Entry(datos, justify=CENTER, textvariable=entrada_usuario).grid(row=0, column=1)

    entrada_contrasenia = IntVar(screen_puntajes)
    Entry(datos, justify=CENTER, textvariable=entrada_contrasenia).grid(row=1, column=1)

    buscar = Button(screen_puntajes, text="Buscar", font=("Courier New", 10), fg="gold", bg="gray22",
                    command=lambda: verificar_perfiles(entrada_usuario, entrada_contrasenia, perfiles, lista_archivos, screen_puntajes))
    buscar.pack()

    salir = Button(screen_puntajes, text="Salir", font=("Courier New", 10), fg="gold", bg="gray22", command=screen_puntajes.destroy)
    salir.pack(pady=20)
```



La función `v_puntajes` crea una ventana para que el usuario ingrese sus datos y así obtener sus debidos puntajes, en caso de que el usuario no exista muestra el mensaje “no hay registros de juego”.

```

def leer_puntajes():
    prueba = open("records_usuario", "r")
    prueba.seek(0)
    lineas_archivo = []
    for linea in prueba:
        lectura = list(linea.rstrip().split(","))
        lineas_archivo.append(lectura)
    prueba.close()
    cont_puntajes = len(lineas_archivo)
    return cont_puntajes, lineas_archivo

```

La función leer\_puntajes lee los registros de los puntajes obtenidos

## TRABAJO A FUTURO

Desde la raíz del programa, se plantea la posibilidad de utilizar diversos métodos de codificación con fin de reducir el numero de líneas y de esta manera optimizar el código, además de hacerlo mas entendible para un tercero; otra opción viable es la P.O.O (Programación Orientada a Objetos). Avanzando un poco mas con la sección de archivos no se cierran las puertas de generar una estadística general por carreras y de esta manera organizar dichos datos con un fin específico.

## CONCLUSIONES

Posterior al desarrollo en la totalidad del proyecto propuesto, se hace evidente la funcionalidad del medio de programación aprendido, ya que con el conocimiento obtenido nos vimos en la capacidad de desarrollar un software que consideramos bastante completo, demostrando como una serie de procesos computacionales desarrollados a lo largo del semestre, nos permitieron llegar a tal punto.

De igual forma, durante el proceso de investigación pudimos evidenciar la cantidad de posibilidades y alcances que tienen los temas aprendidos, ya que sus aplicaciones poseen un alcance enorme capaz de implementarse en prácticamente cualquier ámbito de un individuo, comunidad o inclusive una sociedad.

## BIBLIOGRAFIA

- [1] (2018, 7 mayo). La mitad de los colombianos consume videojuegos, según reporte. *Tendencias de consumo de videojuegos en Colombia 2018*. Tomado de <https://www.dinero.com>
- [2] Lapse: A Forgotten Future Estrategia móvil – Grupo Alfas. (2015, 15 julio). Tomado de <http://grupoalfas.com/lapse-a-forgotten-future-estrategia-movil/>
- [3] TkInter para Python: ¿Cómo hacer una ventana? (Básico). (2017, 14 abril). [Archivo de vídeo]. Tomado de <https://www.youtube.com/watch?v=iPXiZALnZFg>
- [4] Curso Python. Interfaces gráficas I. Vídeo 42. (2018, 31 enero). [Archivo de vídeo]. Tomado de <https://www.youtube.com/watch?v=hTUJC8HsC2I>
- [5] Turtle in Tkinter creating multiple windows. (2018, 22 septiembre). Tomado de <https://stackoverflow.com/questions/52452857/turtle-in-tkinter-creating-multiple-windows>

- [6] Add tkinter's intvar to an integer. (2013, 1 noviembre). Tomado de <https://stackoverflow.com/questions/19719577/add-tkinters-intvar-to-an-integer>
- [7] Python Tkinter: Rightmost column and bottommost row value for grid attribute? (2019, 28 febrero). Tomado de <https://stackoverflow.com/questions/54931797/python-tkinter-rightmost-column-and-bottommost-row-value-for-grid-attribute>
- [8] Integrate turtle module with tkinter canvas. (2017, 20 junio). Tomado de <https://stackoverflow.com/questions/44653500/integrate-turtle-module-with-tkinter-canvas>
- [9] How to pass arguments to a Button command in Tkinter? (2011, 3 agosto). Tomado de <https://stackoverflow.com/questions/6920302/how-to-pass-arguments-to-a-button-command-in-tkinter>
- [10] \_tkinter.TclError: expected floating-point number but got «08» (looks like invalid octal number). (2017, 14 abril). Tomado de <https://stackoverflow.com/questions/43409795/tkinter-tclerror-expected-floating-point-number-but-got-08-looks-like-inval>
- [11] Why it shows blank instead of picture in my Tkinter program? (2013, 28 diciembre). Tomado de <https://stackoverflow.com/questions/20812579/why-it-shows-blank-instead-of-picture-in-my-tkinter-program/20812619>
- [12] random-Generar números pseudoaleatorios. Tomado de <https://docs.python.org/3/library/random.html>
- [13] El Administrador de Geometría de Tkinter Pack. Tomado de <https://effbot.org/tkinterbook/pack.htm>
- [14] Curso Python. Interfaces gráficas I. Vídeo 42. (2018, 31 enero). [Archivo de vídeo]. Tomado de <https://www.youtube.com/watch?v=hTUJC8HsC2I>