

Instituto Ind. Luis. A. Huergo

Análisis de Sistemas

Trabajo Práctico 3er Bimestre

Proyecto Python

Fecha de entrega: plazo hasta el martes 16 de Septiembre de 2024

**Integrantes:** Facundo Benítez, Joaquín Iñaki Piñeiro y Joaquín Hernan Rodriguez

**Tema elegido:** Palabras Huergo(juego inspirado en el juego Pasapalabras)

**Descripción del proyecto:** El juego consiste en que una persona tendrá que responder una serie de preguntas de diferentes tipos como: cultura general, adivinar un personaje, etc. Cada pregunta tendrá 4 opciones, los aciertos sumarán 10 segundos a un contador que será su tiempo en el juego final(el rosco), pero los errores no sumaran nada.

El rosco será una rueda con las letras del abecedario(excepto la K y la W), en la cual deberás adivinar una palabra que empezará o contendrá la letra asignada, en caso de no saber alguna palabra podrás saltearla con un botón de “Pasapalabra”, la cual podrás responder más tarde.

**Problemas “técnicos” encontrados a la hora de pasar de la idea al código:** Los problemas a nuestro pesar fueron demasiados, en primera instancia el tener que programar con bibliotecas completamente nuevas y con enfoques tan diferentes a lo trabajado anteriormente fue un problema y un reto para nosotros. Otro problema fue el ver como lograr hacer un juego con tantas mecánicas como es el pasapalabra, yendo de problemas como lo era saber los diseños que queríamos, las preguntas y las mecánicas las cuales son complicadas de definir, pero sin duda los problemas que mayor dificultad trajeron son 2: la implementación de diferentes librerías y la combinación con nuestro código fue de lo más complejo, principalmente visto en el desarrollo de la rocola y el rosco. El otro fue la organización interna, ya que en nuestro grupo al querer hacer 2 proyectos género que 2 de los integrantes trabajarán en uno y dejaran al otro trabajando solo lo cual a la larga generó muchos problemas.

**Métodos utilizados para llegar a la resolución de dichos inconvenientes:** La forma de la resolución de todos los problemas, de los diseños y mecánicas fue hablarlo tanto con el profe como con el grupo. La investigación fue algo que se fue haciendo a medida que se iba requiriendo aprender una técnica o función nueva. Por parte de los problemas centrales, el desarrollo y la combinación de los códigos de la rocola fue avance de prueba y error, además de reestructurar el código. Por parte del rosco y la organización del grupo aunque sí hay mejoras la verdad es que aún hay muchos problemas que resolver.

**(En el caso de que hayan utilizado chat GPT) Describa ventajas y desventajas de utilizar esta herramienta:** Chat GPT es una herramienta que si implementamos como ayuda o guía en el desarrollo del juego, aunque depende de las instrucciones que le des o de la forma en que se las des. En sus inicios fue de lo más útil para resolver problemas al no ser tan grande nuestro código, además de que fue de mucha ayuda para poder plasmar y explicar ciertas ideas que no sabíamos cómo encarar, pero mientras más complejo se volvían las peticiones a la IA, más complicado se volvía que ayude de la manera correcta, a veces siendo más un problema que un ayuda, y terminando siendo mucho mejor hacerlo por nuestra propia cuenta.

**¿Han trabajado como un grupo realmente?:** Honestamente no podemos decir que trabajamos en grupo, debido a que desde un inicio nos dividimos en 2 grupos, lo cual generó que 2 personas trabajaran juntas y otra trabajara sola la gran mayoría del tiempo, y aunque sería genial decir que así logramos realizar todo correctamente, la realidad es muy diferente. Uno de los proyectos quedó abandonado por la dificultad de este mismo, y el otro por haber sido programado por una persona genera que las otras 2 no sepan cómo ayudar al desarrollo de este programa de manera correcta.

**¿Creen que durante este año han crecido y madurado a la hora de programar?:** A lo largo de este año escolar, fue notorio el crecimiento y la maduración de los integrantes del grupo a la hora de realizar nuevos programas o juegos, pero creemos que podríamos haber aprendido más si nos hubieran dado más tiempo para realizar los proyectos o nos hubieran explicado más detenidamente los temas de la materia, como lo fueron el Pygame, Jason o TKinter.

**¿Han sabido manejar bien los tiempos?:** Consideramos que al tener más de un mes para poder trabajar en el proyecto nos sobraría tiempo, además de no tener que hacerlo completamente solos, por lo que nos tomamos mucho tiempo para el desarrollo del programa, pero a medida que iban transcurriendo los días nos dabamos cuenta de que no teníamos tanto tiempo como hubiéramos querido, cosa que generó la necesidad de tener que acelerar los procesos de desarrollos, tener que trabajar en nuestros hogares y que tuviéramos que trabajar hasta el dia anterior a la entrega del trabajo.

**¿Supieron poner bien los límites hasta dónde querían llegar en el mismo?¿Llegaron al mismo?¿Lo superaron?:** Si logramos establecer los límites y los objetivos que queríamos realizar en el proyecto, siendo que nos pusimos objetivos para ir alcanzando de a poco, logrando de esta manera llegar a estos mismos objetivos, pero lo que los objetivos más grandes como lo era una selección random no fue posible, por lo que no creo que hayamos llegado a cumplir todos los objetivos, pero si a los esenciales.

**¿Sienten que luego de haber tenido tantas horas en el Laboratorio haciendo ejercicios previos, han logrado autonomía y pueden encarar programaciones sin que el profe está todo el tiempo reviendo su código?:** Luego de realizar este proyecto, podemos afirmar que a pesar de necesitar ayuda en ciertas ocasiones, podríamos llegar a realizar otros proyectos o programas de manera autónoma y sin el apoyo del profesor.

**Pegar el código totalmente comentado debajo:**

**(además de subirlo al Classroom con formato 3 AO\_[Apellidos]\_[nombre proyecto] ejemplo: Malvasio Salvia\_TP\_MascotaVirtual)**

| import cv2  import pygame  import sys  from moviepy.editor import VideoFileClip  # Inicializar Pygame y sus módulos  pygame.init()  pygame.mixer.init()  # Definir colores  Gris = (88, 88, 88)  Negro = (0, 0, 0)  Blanco = (255, 255, 255)  Verde = (0, 255, 0)  Rojo = (255, 0, 0)  # Configurar la pantalla  Pantalla = pygame.display.set\_mode((900, 508))  pygame.display.set\_caption("Palabras Huergo")  # Cargar fondos  Fondo\_Menu = pygame.image.load('Fondos/Fondo.jpg')  Fondo\_Tutorial = pygame.image.load('Fondos/Fondo Tutorial.png')  Fondo\_Mini\_Juego\_1 = pygame.image.load('Fondos/Fondo MiniJuegos 1.png')  Fondo\_Mini\_Juego\_2 = pygame.image.load('Fondos/Fondo MiniJuegos 2.png')  Fondo\_Mini\_Juego\_3 = pygame.image.load('Fondos/Fondo MiniJuegos 3.png')  Fondo\_Mini\_Juego\_4 = pygame.image.load('Fondos/Fondo MiniJuegos 4.png')  Fondo\_Mini\_Juego\_5 = pygame.image.load('Fondos/Fondo MiniJuegos 5.png')  Fondo\_inicio = Fondo\_Menu  # Cargar botones  Boton\_diseño = pygame.image.load('Botones/Boton.png')  Boton\_Oscuro = pygame.image.load('Botones/Boton Oscuro.png')  Boton\_return = pygame.image.load('Botones/Boton de Return.png')  Boton\_ret\_o = pygame.image.load('Botones/Boton de Return Oscuro.png')  Boton\_inicio = Boton\_diseño  Boton\_Tutorial = pygame.image.load('Botones/Tutorial\_Boton.png')  Boton\_Tutorial\_Oscuro = pygame.image.load('Botones/Tutorial\_Boton Oscuro.png')  Rectan = Boton\_Tutorial.get\_rect(topleft=(338, 320))  Rectangulo = Boton\_diseño.get\_rect(topleft=(325, 230))  Boton\_Play\_Pausa = pygame.Rect(320, 410, 100, 50)  Boton\_Restart = pygame.Rect(495, 410, 100, 50)  # Definir fuentes  Fuente\_Botones = pygame.font.Font("Tipografias/ShowcardGothic.ttf", 30)  Fuente\_pregunta = pygame.font.Font('Tipografias/cooper-black.ttf', 30)  Fuente\_opcion = pygame.font.Font('Tipografias/cooper-black.ttf', 24)  Fuente\_opcion\_2 = pygame.font.Font('Tipografias/cooper-black.ttf', 35)  Fuente\_opcion\_3 = pygame.font.Font('Tipografias/cooper-black.ttf', 28)  Fuente\_opcion\_4 = pygame.font.Font('Tipografias/cooper-black.ttf', 20)  Fuente\_texto = pygame.font.Font('Tipografias/ShowcardGothic.ttf', 25)  #Definir Imagen  imagenJ = pygame.image.load("Banderas/Japon.png")  imagenJ = pygame.transform.scale(imagenJ, (320, 200))  imagenE = pygame.image.load("Banderas/Ecuador.png")  imagenE = pygame.transform.scale(imagenE, (320, 200))  imagenP = pygame.image.load("Banderas/Pakistan.png")  imagenP = pygame.transform.scale(imagenP, (320, 200))  imagenG = pygame.image.load("Banderas/Grecia.png")  imagenG = pygame.transform.scale(imagenG, (320, 200))  imagenT = pygame.image.load("Banderas/Tailandia.png")  imagenT = pygame.transform.scale(imagenT, (320, 200))  imagene = pygame.image.load("Banderas/Egipto.png")  imagene = pygame.transform.scale(imagene, (320, 200))  imagenA = pygame.image.load("FootBall/Szboloszlai.png")  imagenA = pygame.transform.scale(imagenA, (240, 300))  imagenC = pygame.image.load("FootBall/McGoat.png")  imagenC = pygame.transform.scale(imagenC, (240, 300))  imagenD = pygame.image.load("FootBall/Borhalland.png")  imagenD = pygame.transform.scale(imagenD, (240, 300))  imagenR = pygame.image.load("FootBall/river.png")  imagenR = pygame.transform.scale(imagenR, (240, 300))  imagenL = pygame.image.load("FootBall/douglas.png")  imagenL = pygame.transform.scale(imagenL, (240, 300))  imagenF = pygame.image.load("FootBall/talleres.png")  imagenF = pygame.transform.scale(imagenF, (240, 300))  # Información de preguntas  informacion = [  {"Pregunta": "¿Cuál es el nombre de la canción?",  "Opcion": ['Mente en Blanco', 'Caos', 'Extra', 'Sour Candy'],  "Correcta": 0,  "Posicion\_Pre": (141, 55),  "Posicion": [(18, 210), (755, 210), (75, 402), (705, 402)]  },  {"Pregunta": "¿Cuál es el nombre de la canción?",  "Opcion": ['Judas', 'Poker Face', 'Bad Romance', 'Blody Marry'],  "Correcta": 1,  "Posicion\_Pre": (141, 55),  "Posicion": [(72, 210), (712, 210), (22, 402), (700, 402)]  },  {"Pregunta": "¿Cuál es el nombre de la canción?",  "Opcion": ['Deja Vu', 'Loba', 'Acrosito', 'Objection'],  "Correcta": 2,  "Posicion\_Pre": (141, 55),  "Posicion": [(60, 210), (748, 210), (58, 402), (715, 402)]  },  {"Pregunta": "¿Cuál es el nombre de la canción?",  "Opcion": ['Man', 'Shake it Up', 'Bad Blood', 'Cruel Summer'],  "Correcta": 3,  "Posicion\_Pre": (141, 55),  "Posicion": [(83, 210), (705, 210), (45, 402), (700, 406)]  },  {"Pregunta": "¿Cuál es el nombre de la canción?",  "Opcion": ['Dance the Night', 'Leviatating', 'Houdini', 'Sweet Pie'],  "Correcta": 0,  "Posicion\_Pre": (141, 55),  "Posicion": [(18, 210), (707, 210), (58, 402), (720, 402)]  },  {"Pregunta": "¿Cuál es el nombre de la canción?",  "Opcion": ['Taste', 'Please', 'Espresso', 'Feathers'],  "Correcta": 0,  "Posicion\_Pre": (141, 55),  "Posicion": [(80, 210), (743, 210), (58, 402), (725, 402)]  },  {"Pregunta": "¿Cuál es la capital de Italia?",  "Opcion": ["A) Paris", "B) Roma", "C) Venecia", "D) Napoli"],  "Correcta": 1,  "Posicion\_Pre": (240,280),  "Posicion": [(180, 370), (480, 370), (180, 438), (480, 438)]  },  {"Pregunta": "¿Cuánto tarda la luz del sol en llegar a la Tierra?",  "Opcion": ['A) 6 minutos', 'B) 8 segundos', 'C) 8 minutos', 'D) 6 segundos'],  "Correcta": 2,  "Posicion\_Pre": (85,280),  "Posicion": [(180, 370), (480, 370), (180, 438), (480, 438)]  },  {"Pregunta": "¿Cuál es el país más grande y el más pequeño?",  "Opcion": ['A) Rusia y Vaticano', 'B) Rusia y China', 'C) EEUU y China', 'D) Rusia y EEUU'],  "Correcta": 0,  "Posicion\_Pre": (100,280),  "Posicion": [(180, 370), (480, 370), (180, 438), (480, 438)]  },  {"Pregunta": "¿Cuál palabra está escrita correctamente?",  "Opcion": ['A) Conclullente', 'B) Mayor', 'C) Embaruyar', 'D) Soslalio'],  "Correcta": 1,  "Posicion\_Pre": (130,280),  "Posicion": [(180, 370), (480, 370), (180, 438), (480, 438)]  },  {"Pregunta": "¿Cuál de estas palabras se escribe con H?",  "Opcion": ['A) Humillacion', 'B) Hiluso', 'C) Hinflar', 'D) Hinodoro'],  "Correcta": 0,  "Posicion\_Pre": (140,280),  "Posicion": [(180, 370), (480, 370), (180, 438), (480, 438)]  },  {"Pregunta": "¿Cuál de las palabras está bien escrita?",  "Opcion": ['A) Mahonesa', 'B) Royo', 'C) Fayar', 'D) Hierba'],  "Correcta": 3,  "Posicion\_Pre": (140,280),  "Posicion": [(180, 370), (480, 370), (180, 438), (480, 438)]  },  {"Pregunta": "¿Adivina quien es el Personaje?",  "Opcion": ['Galileo Galilei', 'Isaac Newton', 'Albert Einstein ', 'Nikola Tesla'],  "Correcta": 3,  "imagen": pygame.image.load('Personaje/Tesla.jpg'),  "Posicion\_Pre": (155,55),  "imagen\_posi": (100,178),  "Posicion": [(445, 225), (692, 225), (438, 405), (697, 405)]  },  {"Pregunta": "¿Adivina quien es el Personaje?",  "Opcion": ['Mirtha Legrand', 'Moria Casan', 'Susana Gimenez', 'Lizy Tagliani'],  "Correcta": 0,  "imagen": pygame.image.load('Personaje/Mirtha Legrand.png'),  "Posicion\_Pre": (155,55),  "imagen\_posi": (113,178),  "Posicion": [(438, 225), (700, 225), (433, 405), (696, 405)]  },  {"Pregunta": "¿Adivina quien es el Personaje?",  "Opcion": ['Hera', 'Afrodita', 'Atenea', 'Demeter'],  "Correcta": 2,  "imagen": pygame.image.load('Personaje/Atenea.jpg'),  "Posicion\_Pre": (155,55),  "imagen\_posi": (110,178),  "Posicion": [(490, 220), (698, 220), (470, 400), (700, 400)]  },  {"Pregunta": "¿A que Juego Pertenece?",  "Opcion": ['EarthBound', 'Deltarune', 'Undertale', 'Final Fantasy'],  "Correcta": 2,  "imagen": pygame.image.load('Personaje/Gaster.png'),  "Posicion\_Pre": (200,55),  "imagen\_posi": (95,178),  "Posicion": [(458, 225), (715, 225), (468, 405), (692, 405)]  },  {"Pregunta": "¿A que Juego Pertenece?",  "Opcion": ['Ratchet & Clank', 'Bioumutant', 'Star Fox', 'Crash Bandicoot'],  "Correcta": 3,  "imagen": pygame.image.load('Personaje/Doctor Neo.png'),  "Posicion\_Pre": (200,55),  "imagen\_posi": (105,178),  "Posicion": [(432, 225), (700, 225), (482, 405), (674, 405)]  },  {"Pregunta": "¿A que Juego Pertenece?",  "Opcion": ['Mortal Kombat', 'Soulcalibur', 'Street Fighter', 'Injustice 2'],  "Correcta": 1,  "imagen": pygame.image.load('Personaje/Mitsurugi.png'),  "Posicion\_Pre": (200,55),  "imagen\_posi": (95,178),  "Posicion": [(440, 225), (705, 225), (445, 405), (710, 405)]  },  {"Pregunta": "¿Cual es la pelicula?",  "Opcion": ['Spiderman 1', 'The Amazing Spiderman', 'Spiderman no way home', 'Spiderman 2'],  "Correcta": 2,  "imagen 1": pygame.image.load('Emoji/Araña.png'),  "imagen 2": pygame.image.load('Emoji/Araña.png'),  "imagen 3": pygame.image.load('Emoji/Araña.png'),  "Posicion\_Pre": (140,280),  "imagen\_posi 1": (140,90),  "imagen\_posi 2": (325,94),  "imagen\_posi 3": (505,94),  "Posicion": [(220, 370), (460, 370), (160, 438), (520, 438)]  },  {"Pregunta": "¿Cual es la pelicula?",  "Opcion": ['Harry Potter', 'Shazam', 'Electra', "Flash"],  "Correcta": 0,  "imagen 1": pygame.image.load('Emoji/Nene.png'),  "imagen 2": pygame.image.load('Emoji/Lentes.png'),  "imagen 3": pygame.image.load('Emoji/Rayo.png'),  "Posicion\_Pre": (140,280),  "imagen\_posi 1": (140,90),  "imagen\_posi 2": (325,94),  "imagen\_posi 3": (505,94),  "Posicion": [(220, 370), (460, 370), (160, 438), (520, 438)]  },  {"Pregunta": "¿Cual es la pelicula?",  "Opcion": ['Jungle', 'Zootopia', 'Balto', 'Madagascar'],  "Correcta": 1,  "imagen 1": pygame.image.load('Emoji/Zorro.png'),  "imagen 2": pygame.image.load('Emoji/Conejo.png'),  "imagen 3": pygame.image.load('Emoji/Placa.png'),  "Posicion\_Pre": (140,280),  "imagen\_posi 1": (140,90),  "imagen\_posi 2": (325,94),  "imagen\_posi 3": (505,94),  "Posicion": [(220, 370), (460, 370), (160, 438), (520, 438)]  },  {"Pregunta": "¿Cual es la pelicula?",  "Opcion": ['It', 'Batman', 'Terrifier', 'Joker'],  "Correcta": 3,  "imagen 1": pygame.image.load('Emoji/Payaso.png'),  "imagen 2": pygame.image.load('Emoji/Pistola.png'),  "imagen 3": pygame.image.load('Emoji/TV.png'),  "Posicion\_Pre": (140,280),  "imagen\_posi 1": (140,90),  "imagen\_posi 2": (325,94),  "imagen\_posi 3": (505,94),  "Posicion": [(220, 370), (460, 370), (160, 438), (520, 438)]  },  {"Pregunta": "¿Cual es la pelicula?",  "Opcion": ['Venom', 'Morbius', 'T-Rex', 'Madam Web'],  "Correcta": 0,  "imagen 1": pygame.image.load('Emoji/Serpiente.png'),  "imagen 2": pygame.image.load('Emoji/Araña.png'),  "imagen 3": pygame.image.load('Emoji/Negro.png'),  "Posicion\_Pre": (140,280),  "imagen\_posi 1": (140,90),  "imagen\_posi 2": (325,94),  "imagen\_posi 3": (505,94),  "Posicion": [(220, 370), (460, 370), (160, 438), (520, 438)]  },  {"Pregunta": "¿Cual es la pelicula?",  "Opcion": ['Venom', 'Morbius', 'Avatar', 'Madam Web'],  "Correcta": 2,  "imagen 1": pygame.image.load('Emoji/Azul.png'),  "imagen 2": pygame.image.load('Emoji/Planeta.png'),  "imagen 3": pygame.image.load('Emoji/Arco.png'),  "Posicion\_Pre": (140,280),  "imagen\_posi 1": (140,90),  "imagen\_posi 2": (325,94),  "imagen\_posi 3": (505,94),  "Posicion": [(220, 370), (460, 370), (160, 438), (520, 438)]  },  {"Pregunta": "¿Cuál es esta bandera?",  "Opcion": ["China", "Corea del Sur", "Bangladesh", "Japón"],  "Correcta": 3,  "imagen": imagenJ,  "Posicion\_Pre": (200,55),  "imagen\_posi": (45,228),  "Posicion": [(496, 228), (692, 228), (464, 394), (733, 394)],  },  {"Pregunta": "¿Cuál es esta bandera?",  "Opcion": ["Venezuela", "Colombia", "Ecuador", "Bolivia"],  "Correcta": 2,  "imagen": imagenE,  "Posicion\_Pre": (217,55),  "imagen\_posi": (45,228),  "Posicion": [(470, 228), (718, 228), (480, 394), (730, 394)],  },  {"Pregunta": "¿Cuál es esta bandera?",  "Opcion": ["Turkeministan", "Argelia", "Pakistan", "Turquia"],  "Correcta": 2,  "imagen": imagenP,  "Posicion\_Pre": (217,55),  "imagen\_posi": (45,228),  "Posicion": [(440, 228), (728, 228), (481, 394), (728, 394)],  },  {"Pregunta": "¿Cuál es esta bandera?",  "Opcion": ["Honduras", "Grecia", "Uruguay", "Guatemala"],  "Correcta": 1,  "imagen": imagenG,  "Posicion\_Pre": (217,55),  "imagen\_posi": (45,228),  "Posicion": [(469, 228), (733, 228), (479, 394), (712, 394)],  },  {"Pregunta": "¿Cuál es esta bandera?",  "Opcion": ["Costa Rica", "Cuba", "Tailandia", "Croacia"],  "Correcta": 2,  "imagen": imagenT,  "Posicion\_Pre": (217,55),  "imagen\_posi": (45,228),  "Posicion": [(467, 228), (743, 228), (475, 394), (716, 394)],  },  {"Pregunta": "¿Cuál es esta bandera?",  "Opcion": ["Yemen", "Egipto", "Irak", "Siria"],  "Correcta": 1,  "imagen": imagene,  "Posicion\_Pre": (217,55),  "imagen\_posi": (45,228),  "Posicion": [(496, 228), (743, 228), (508, 394), (752, 394)],  },  {"Pregunta": "¿ Cuál es este jugador?",  "Opcion": ["Xhaka", "Wirtz", "Szoboszlai", "Bellingham"],  "Correcta": 2,  "imagen": imagenA,  "Posicion\_Pre": (220,55),  "imagen\_posi": (95,180),  "Posicion": [(488, 230), (735, 230), (468, 405), (708, 405)],  },  {"Pregunta": "¿ Cuál es este jugador?",  "Opcion": ["Mac Allister", "McTominay", "Mcginn", "Veretout"],  "Correcta": 1,  "imagen": imagenC,  "Posicion\_Pre": (220,55),  "imagen\_posi": (95,180),  "Posicion": [(460, 228), (708, 228), (483, 405), (722, 405)],  },  {"Pregunta": "¿ Cuál es este jugador?",  "Opcion": ["De la cruz", "Pratto", "Paulo Diaz", "Borhalland"],  "Correcta": 3,  "imagen": imagenD,  "Posicion\_Pre": (220,55),  "imagen\_posi": (95,180),  "Posicion": [(472, 228), (735, 228), (468, 405), (709, 405)],  },  {"Pregunta": "¿ Cuál es este equipo?",  "Opcion": ["Nacional Potosí", "River Plate", "River Plate(URU)", "Boca"],  "Correcta": 0,  "imagen": imagenR,  "Posicion\_Pre": (220,55),  "imagen\_posi": (95,180),  "Posicion": [(440, 228), (710, 228), (423, 405), (745, 405)],  },  {"Pregunta": "¿ Cuál es este equipo?",  "Opcion": ["Newells", "Douglas", "Chacarita", "Colon"],  "Correcta": 1,  "imagen": imagenL,  "Posicion\_Pre": (220,55),  "imagen\_posi": (95,180),  "Posicion": [(485, 228), (728, 228), (472, 405), (738, 405)],  },  {"Pregunta": "¿ Cuál es este equipo?",  "Opcion": ["Barracas Central", "San Martín (T)", "Talleres", "Internacional"],  "Correcta": 2,  "imagen": imagenF,  "Posicion\_Pre": (220,55),  "imagen\_posi": (95,180),  "Posicion": [(430, 228), (685, 228), (485, 405), (690, 405)],  }  ]  # Variables globales  Pregunta\_actual = 0  Opcion\_sel = 0  Tiempo = 0  respuesta\_correcta = None  Jugando = False  Pausado = False  Video = None  Ruta\_Video = None  Audio = None  # Función para cargar audio y video  def cargar\_audio\_y\_video(Pregunta\_actual):  global Video, Ruta\_Video, Audio  if Fondo\_inicio == Fondo\_Mini\_Juego\_3:  if Pregunta\_actual == 0:  Audio = "Canciones/Mente en Blanco.mp3"  Ruta\_Video = "Canciones/Ondas.mp4"  elif Pregunta\_actual == 1:  Audio = "Canciones/Poker Face.mp3"  Ruta\_Video = "Canciones/Ondas.mp4"  elif Pregunta\_actual == 2:  Audio = "Canciones/Acróstico.mp3"  Ruta\_Video = "Canciones/Ondas.mp4"  elif Pregunta\_actual == 3:  Audio = "Canciones/Cruel Summer.mp3"  Ruta\_Video = "Canciones/Ondas.mp4"  elif Pregunta\_actual == 4:  Audio = "Canciones/Dance the Night.mp3"  Ruta\_Video = "Canciones/Ondas.mp4"  elif Pregunta\_actual == 5:  Audio = "Canciones/Taste.mp3"  Ruta\_Video = "Canciones/Ondas.mp4"  else:  Audio = "Canciones/Fondo.mp3"  Ruta\_Video = None  Video = cv2.VideoCapture(Ruta\_Video)  else:  Audio = "Canciones/Sin Sonido.mp3"  # Función para reproducir la música  def reproducir\_musica():  pygame.mixer.music.load(Audio)  pygame.mixer.music.play(-1)  # Función para detener la música  def detener\_musica():  pygame.mixer.music.stop()  # Función para dibujar botones  def Botones(surface, Texto, rect):  mouse\_pos = pygame.mouse.get\_pos()  if rect.collidepoint(mouse\_pos):  pygame.draw.rect(surface, Gris, rect)  texto\_color = Negro  else:  pygame.draw.rect(surface, Gris, rect)  texto\_color = Blanco  texto = Fuente\_Botones.render(Texto, True, texto\_color)  surface.blit(texto, (rect.x + (rect.width - texto.get\_width()) // 2,  rect.y + (rect.height - texto.getheight()) // 2))    # Función para dibujar el botón de inicio  def Boton\_i(surface):  Posi\_mouse = pygame.mouse.get\_pos()  if Fondo\_inicio == Fondo\_Menu or Fondo\_inicio == Fondo\_Mini\_Juego\_1 or Fondo\_inicio == Fondo\_Mini\_Juego\_2 or Fondo\_inicio == Fondo\_Mini\_Juego\_3:  if Fondo\_inicio == Fondo\_Menu:  if Rectangulo.collidepoint(Posi\_mouse):  surface.blit(Boton\_Oscuro, Rectangulo.topleft)  else:  surface.blit(Boton\_diseño, Rectangulo.topleft)  elif Fondo\_inicio == Fondo\_Mini\_Juego\_1:  if Rectangulo.collidepoint(Posi\_mouse):  surface.blit(Boton\_ret\_o, Rectangulo.topleft)  else:  surface.blit(Boton\_return, Rectangulo.topleft)  # Función para dibujar el botón de Tutorial  def Boton\_T(surface):  Posi\_mouse = pygame.mouse.get\_pos()  if Fondo\_inicio == Fondo\_Menu or Fondo\_inicio == Fondo\_Tutorial:  if Fondo\_inicio == Fondo\_Menu:  if Rectan.collidepoint(Posi\_mouse):  surface.blit(Boton\_Tutorial\_Oscuro, Rectan.topleft)  else:  surface.blit(Boton\_Tutorial, Rectan.topleft)  elif Fondo\_inicio == Fondo\_Tutorial:  if Rectan.collidepoint(Posi\_mouse):  surface.blit(Boton\_ret\_o, Rectan.topleft)  else:  surface.blit(Boton\_return, Rectan.topleft)  else:  None  # Función para dibujar el mini juego  def Mini\_Juego(informacion\_pregunta, tiempo):  if Pregunta\_actual <= 5:  Pantalla.blit(Fondo\_Mini\_Juego\_3,(0, 0))  Pregunta\_text = Fuente\_pregunta.render(informacion\_pregunta["Pregunta"], True, Blanco)  Pantalla.blit(Pregunta\_text, informacion\_pregunta["Posicion\_Pre"])  for i, option in enumerate(informacion\_pregunta["Opcion"]):  x, y = informacion\_pregunta["Posicion"][i]  Usar = Fuente\_opcion if i == 0 else Fuente\_opcion\_3  Usar\_3 = Fuente\_opcion if i == 3 else Fuente\_opcion\_3  if Pregunta\_actual == 0 or Pregunta\_actual == 4:  if respuesta\_correcta is not None:  if i == informacion\_pregunta["Correcta"]:  text = Usar.render(option, True, Verde)  elif i == Opcion\_sel and not respuesta\_correcta:  text = Usar.render(option, True, Rojo)  else:  text = Usar.render(option, True, Blanco)  else:  if i == Opcion\_sel:  text = Usar.render(option, True, Negro)  else:  text = Usar.render(option, True, Blanco)  Pantalla.blit(text, (x, y))  elif Pregunta\_actual == 3:  if respuesta\_correcta is not None:  if i == informacion\_pregunta["Correcta"]:  text = Usar\_3.render(option, True, Verde)  elif i == Opcion\_sel and not respuesta\_correcta:  text = Usar\_3.render(option, True, Rojo)  else:  text = Usar\_3.render(option, True, Blanco)  else:  if i == Opcion\_sel:  text = Usar\_3.render(option, True, Negro)  else:  text = Usar\_3.render(option, True, Blanco)  Pantalla.blit(text, (x, y))  else:  if respuesta\_correcta is not None:  if i == informacion\_pregunta["Correcta"]:  text = Fuente\_opcion\_3.render(option, True, Verde)  elif i == Opcion\_sel and not respuesta\_correcta:  text = Fuente\_opcion\_3.render(option, True, Rojo)  else:  text = Fuente\_opcion\_3.render(option, True, Blanco)  else:  if i == Opcion\_sel:  text = Fuente\_opcion\_3.render(option, True, Negro)  else:  text = Fuente\_opcion\_3.render(option, True, Blanco)  Pantalla.blit(text, (x, y))  Texto = Fuente\_texto.render(str(int(tiempo)), True, Blanco)  Pantalla.blit(Texto, (815, 47))  elif Pregunta\_actual <= 11:  Pantalla.blit(Fondo\_Mini\_Juego\_1, (0, 0))  Pregunta\_text = Fuente\_pregunta.render(informacion\_pregunta["Pregunta"], True, Blanco)  Pantalla.blit(Pregunta\_text, informacion\_pregunta["Posicion\_Pre"])  for i, option in enumerate(informacion\_pregunta["Opcion"]):  x, y = informacion\_pregunta["Posicion"][i]  if respuesta\_correcta is not None:  if i == informacion\_pregunta["Correcta"]:  text = Fuente\_opcion.render(option, True, Verde)  elif i == Opcion\_sel and not respuesta\_correcta:  text = Fuente\_opcion.render(option, True, Rojo)  else:  text = Fuente\_opcion.render(option, True, Blanco)  else:  if i == Opcion\_sel:  text = Fuente\_opcion.render(option, True, Negro)  else:  text = Fuente\_opcion.render(option, True, Blanco)  Pantalla.blit(text, (x, y))  Texto = Fuente\_texto.render(str(tiempo), True, Blanco)  Pantalla.blit(Texto, (815, 47))  elif Pregunta\_actual <= 17:  Pantalla.blit(Fondo\_Mini\_Juego\_2, (0, 0))  Pregunta\_text = Fuente\_pregunta.render(informacion\_pregunta["Pregunta"], True, Blanco)  Pantalla.blit(Pregunta\_text, informacion\_pregunta["Posicion\_Pre"])  Pantalla.blit(informacion\_pregunta["imagen"], informacion\_pregunta["imagen\_posi"])  for i, option in enumerate(informacion\_pregunta["Opcion"]):  x, y = informacion\_pregunta["Posicion"][i]  if Pregunta\_actual == 14:  if respuesta\_correcta is not None:  if i == informacion\_pregunta["Correcta"]:  text = Fuente\_opcion\_2.render(option, True, Verde)  elif i == Opcion\_sel and not respuesta\_correcta:  text = Fuente\_opcion\_2.render(option, True, Rojo)  else:  text = Fuente\_opcion\_2.render(option, True, Blanco)  else:  if i == Opcion\_sel:  text = Fuente\_opcion\_2.render(option, True, Negro)  else:  text = Fuente\_opcion\_2.render(option, True, Blanco)  Pantalla.blit(text, (x, y))  else:  if respuesta\_correcta is not None:  if i == informacion\_pregunta["Correcta"]:  text = Fuente\_opcion.render(option, True, Verde)  elif i == Opcion\_sel and not respuesta\_correcta:  text = Fuente\_opcion.render(option, True, Rojo)  else:  text = Fuente\_opcion.render(option, True, Blanco)  else:  if i == Opcion\_sel:  text = Fuente\_opcion.render(option, True, Negro)  else:  text = Fuente\_opcion.render(option, True, Blanco)  Pantalla.blit(text, (x, y))  Texto = Fuente\_texto.render(str(tiempo), True, Blanco)  Pantalla.blit(Texto, (815, 47))  elif Pregunta\_actual <= 23:  Pantalla.blit(Fondo\_Mini\_Juego\_4, (0, 0))  Pregunta\_text = Fuente\_pregunta.render(informacion\_pregunta["Pregunta"], True, Blanco)  Pantalla.blit(Pregunta\_text, informacion\_pregunta["Posicion\_Pre"])  Pantalla.blit(informacion\_pregunta["imagen 1"], informacion\_pregunta["imagen\_posi 1"])  Pantalla.blit(informacion\_pregunta["imagen 2"], informacion\_pregunta["imagen\_posi 2"])  Pantalla.blit(informacion\_pregunta["imagen 3"], informacion\_pregunta["imagen\_posi 3"])  for i, option in enumerate(informacion\_pregunta["Opcion"]):  x, y = informacion\_pregunta["Posicion"][i]  if respuesta\_correcta is not None:  if i == informacion\_pregunta["Correcta"]:  text = Fuente\_opcion\_4.render(option, True, Verde)  elif i == Opcion\_sel and not respuesta\_correcta:  text = Fuente\_opcion\_4.render(option, True, Rojo)  else:  text = Fuente\_opcion\_4.render(option, True, Blanco)  else:  if i == Opcion\_sel:  text = Fuente\_opcion\_4.render(option, True, Negro)  else:  text = Fuente\_opcion\_4.render(option, True, Blanco)  Pantalla.blit(text, (x, y))  Texto = Fuente\_texto.render(str(tiempo), True, Blanco)  Pantalla.blit(Texto, (815, 47))  elif Pregunta\_actual <= 29:  Pantalla.blit(Fondo\_Mini\_Juego\_5, (0, 0))  Pregunta\_text = Fuente\_pregunta.render(informacion\_pregunta["Pregunta"], True, Blanco)  Pantalla.blit(Pregunta\_text, informacion\_pregunta["Posicion\_Pre"])  Pantalla.blit(informacion\_pregunta["imagen"], informacion\_pregunta["imagen\_posi"])  for i, option in enumerate(informacion\_pregunta["Opcion"]):  x, y = informacion\_pregunta["Posicion"][i]  if respuesta\_correcta is not None:  if i == informacion\_pregunta["Correcta"]:  text = Fuente\_opcion.render(option, True, Verde)  elif i == Opcion\_sel and not respuesta\_correcta:  text = Fuente\_opcion.render(option, True, Rojo)  else:  text = Fuente\_opcion.render(option, True, Blanco)  else:  if i == Opcion\_sel:  text = Fuente\_opcion.render(option, True, Negro)  else:  text = Fuente\_opcion.render(option, True, Blanco)  Pantalla.blit(text, (x, y))  Texto = Fuente\_texto.render(str(tiempo), True, Blanco)  Pantalla.blit(Texto, (815, 47))  else:  Pantalla.blit(Fondo\_Mini\_Juego\_2, (0, 0))  Pregunta\_text = Fuente\_pregunta.render(informacion\_pregunta["Pregunta"], True, Blanco)  Pantalla.blit(Pregunta\_text, informacion\_pregunta["Posicion\_Pre"])  Pantalla.blit(informacion\_pregunta["imagen"], informacion\_pregunta["imagen\_posi"])  for i, option in enumerate(informacion\_pregunta["Opcion"]):  x, y = informacion\_pregunta["Posicion"][i]  if respuesta\_correcta is not None:  if i == informacion\_pregunta["Correcta"]:  text = Fuente\_opcion.render(option, True, Verde)  elif i == Opcion\_sel and not respuesta\_correcta:  text = Fuente\_opcion.render(option, True, Rojo)  else:  text = Fuente\_opcion.render(option, True, Blanco)  else:  if i == Opcion\_sel:  text = Fuente\_opcion.render(option, True, Negro)  else:  text = Fuente\_opcion.render(option, True, Blanco)  Pantalla.blit(text, (x, y))  Texto = Fuente\_texto.render(str(tiempo), True, Blanco)  Pantalla.blit(Texto, (815, 47))    # Bucle principal  while True:  Reloj = pygame.time.Clock()  for event in pygame.event.get():  if event.type == pygame.QUIT:  pygame.quit()  sys.exit()  elif event.type == pygame.MOUSEBUTTONDOWN:  mouse\_pos = pygame.mouse.get\_pos()  if Rectangulo.collidepoint(mouse\_pos):  if Fondo\_inicio == Fondo\_Menu:  if Pregunta\_actual <= 5:  Fondo\_inicio = Fondo\_Mini\_Juego\_3  Boton\_inicio = Boton\_return  Rectangulo = Boton\_return.get\_rect(topleft=(10, 15))  Rectan = Boton\_ret\_o.get\_rect(topleft=(900, 15))  if not Jugando:  cargar\_audio\_y\_video(Pregunta\_actual)  reproducir\_musica()  Jugando = True  Pausado = False  elif Pregunta\_actual <= 11:  Fondo\_inicio = Fondo\_Mini\_Juego\_1  Boton\_inicio = Boton\_return  Rectangulo = Boton\_return.get\_rect(topleft=(10, 15))  Rectan = Boton\_ret\_o.get\_rect(topleft=(900, 15))  if Video:  Video.release()  Video = None  Jugando = False  Pausado = False  elif Pregunta\_actual <= 17:  Fondo\_inicio = Fondo\_Mini\_Juego\_2  Boton\_inicio = Boton\_return  Rectangulo = Boton\_return.get\_rect(topleft=(10, 15))  Rectan = Boton\_ret\_o.get\_rect(topleft=(900, 15))  if Video:  Video.release()  Video = None  Jugando = False  Pausado = False  elif Pregunta\_actual <= 23:  Fondo\_inicio = Fondo\_Mini\_Juego\_4  Boton\_inicio = Boton\_return  Rectangulo = Boton\_return.get\_rect(topleft=(10, 15))  Rectan = Boton\_ret\_o.get\_rect(topleft=(900, 15))  if Video:  Video.release()  Video = None  Jugando = False  Pausado = False  elif Pregunta\_actual <= 29:  Fondo\_inicio = Fondo\_Mini\_Juego\_5  Boton\_inicio = Boton\_return  Rectangulo = Boton\_return.get\_rect(topleft=(10, 15))  Rectan = Boton\_ret\_o.get\_rect(topleft=(900, 15))  if Video:  Video.release()  Video = None  Jugando = False  Pausado = False  else:  Fondo\_inicio = Fondo\_Mini\_Juego\_2  Boton\_inicio = Boton\_return  Rectangulo = Boton\_return.get\_rect(topleft=(10, 15))  Rectan = Boton\_ret\_o.get\_rect(topleft=(900, 15))  if Video:  Video.release()  Video = None  Jugando = False  Pausado = False  else:  Fondo\_inicio = Fondo\_Menu  Boton\_inicio = Boton\_diseño  Rectangulo = Boton\_diseño.get\_rect(topleft=(325, 230))  Rectan = Boton\_Tutorial.get\_rect(topleft=(338, 320))  if Video:  Video.release()  Video = None  detener\_musica()  Jugando = False  Pausado = False  if Rectan.collidepoint(event.pos):  if Fondo\_inicio == Fondo\_Menu:  Fondo\_inicio = Fondo\_Tutorial  Boton\_inicio = Boton\_return  Rectan = Boton\_return.get\_rect(topleft=(10, 15))  if Video:  Video.release()  Video = None  Jugando = False  Pausado = False  else:  Fondo\_inicio = Fondo\_Menu  Boton\_inicio = Boton\_Tutorial  Rectan = Boton\_Tutorial.get\_rect(topleft=(338, 320))  if Video:  Video.release()  Video = None  detener\_musica()  Jugando = False  Pausado = False  elif Jugando and event.type == pygame.MOUSEBUTTONDOWN:  if Boton\_Play\_Pausa.collidepoint(mouse\_pos):  if Pausado:  pygame.mixer.music.unpause()  Pausado = False  else:  pygame.mixer.music.pause()  Pausado = True  elif Boton\_Restart.collidepoint(mouse\_pos):  detener\_musica()  cargar\_audio\_y\_video(Pregunta\_actual)  reproducir\_musica()  Video.set(cv2.CAP\_PROP\_POS\_FRAMES, 0)  Pausado = False  elif event.type == pygame.KEYDOWN:  if Jugando:  if event.key == pygame.K\_LEFT:  Opcion\_sel = (Opcion\_sel - 1) % len(informacion[Pregunta\_actual]["Opcion"])  respuesta\_correcta = None  elif event.key == pygame.K\_RIGHT:  Opcion\_sel = (Opcion\_sel + 1) % len(informacion[Pregunta\_actual]["Opcion"])  respuesta\_correcta = None  elif event.key == pygame.K\_UP:  Opcion\_sel = (Opcion\_sel + 2) % len(informacion[Pregunta\_actual]["Opcion"])  respuesta\_correcta = None  elif event.key == pygame.K\_DOWN:  Opcion\_sel = (Opcion\_sel - 2) % len(informacion[Pregunta\_actual]["Opcion"])  respuesta\_correcta = None  elif event.key == pygame.K\_RETURN:  if respuesta\_correcta is None:  if Opcion\_sel == informacion[Pregunta\_actual]["Correcta"]:  Tiempo += 10  respuesta\_correcta = True  else:  respuesta\_correcta = False  if Pregunta\_actual <= 5:  Mini\_Juego(informacion[Pregunta\_actual], Tiempo)  Botones(Pantalla, 'Play' if not Jugando else 'Pause' if not Pausado  else 'Resume', Boton\_Play\_Pausa)  Botones(Pantalla, 'Restart', Boton\_Restart)  Boton\_i(Pantalla)  pygame.display.flip()  pygame.time.delay(500)  Pregunta\_actual += 1  Opcion\_sel = 0  respuesta\_correcta = None  if Pregunta\_actual >= len(informacion):  Pregunta\_actual = 0  Opcion\_sel = 0  Tiempo = 0  Fondo\_inicio = Fondo\_Menu  Boton\_inicio = Boton\_diseño  Rectangulo = Boton\_diseño.get\_rect(topleft=(325, 250))  if Video:  Video.release()  Video = None  detener\_musica()  Jugando = False  Pausado = False  cargar\_audio\_y\_video(Pregunta\_actual)  reproducir\_musica()  else:  Mini\_Juego(informacion[Pregunta\_actual], Tiempo)  Boton\_i(Pantalla)  pygame.display.flip()  pygame.time.delay(500)  Pregunta\_actual += 1  Opcion\_sel = 0  respuesta\_correcta = None  if Pregunta\_actual >= len(informacion):  Pregunta\_actual = 0  Opcion\_sel = 0  Tiempo = 0  Fondo\_inicio = Fondo\_Menu  Boton\_inicio = Boton\_diseño  Rectangulo = Boton\_diseño.get\_rect(topleft=(325, 230))  Rectan = Boton\_Tutorial.get\_rect(topleft=(338, 320))  Jugando = False  Pausado = False  detener\_musica()  pygame.time.delay(500)  Pantalla.blit(Fondo\_inicio, (0, 0))  if Fondo\_inicio != Fondo\_Menu:  if Fondo\_inicio == Fondo\_Mini\_Juego\_3:  Boton\_i(Pantalla)  Mini\_Juego(informacion[Pregunta\_actual], Tiempo)  if Video is not None and Jugando and not Pausado:  ret, frame = Video.read()  if ret:  frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)  frame = cv2.transpose(frame)  frame = cv2.flip(frame, flipCode=1)  frame\_surface = pygame.surfarray.make\_surface(frame)  frame\_surface = pygame.transform.scale(frame\_surface, (413, 217))  Pantalla.blit(frame\_surface, (245, 153))  pygame.draw.rect(Pantalla, Negro, (558, 321, 100, 50))  else:  Video.set(cv2.CAP\_PROP\_POS\_FRAMES, 0)  detener\_musica()  if Pregunta\_actual <= 5:  Botones(Pantalla, 'Play' if not Jugando else 'Pause' if not Pausado else 'Resume', Boton\_Play\_Pausa)  Botones(Pantalla, 'Restart', Boton\_Restart)  Reloj.tick(30)  elif Fondo\_inicio == Fondo\_Tutorial:  Boton\_T(Pantalla)  Pregunta\_actual = 0  Opcion\_sel = 0  Tiempo = 0  else:  Pantalla.blit(Fondo\_inicio, (0, 0))  Boton\_T(Pantalla)  Boton\_i(Pantalla)  Pregunta\_actual = 0  Opcion\_sel = 0  Tiempo = 0  pygame.display.update()  pygame.display.flip() |
| --- |

**Listar y explicar todas las funciones y librerías asociadas que utilizaron. Ya sea personalizadas del estilo de “alimentar mascota(ali 1, ali2):” o que posee el lenguaje “def sort(lista):”**

* **Llamado de las librerías:** Aquí llamó a las diferentes bibliotecas usadas durante el desarrollo del juego, las cuales son cv2, pygame, sys, de movimoviepy.editor el VideoFileClip
* **Carga:** Iniciamos el pygame y el mixer.
* **Definición de los fondos:** Se cargan los diseños de los diferentes fondos.
* **Definición del fondo inicio:** Se define la variable de fondo inicio que será el fondo que queremos ver.
* **Definición de los botones:** Se cargan los diseños de los Botones fondos.
* **Definición del botón inicio:** Se define la variable de Boton inicio que será el botón que queremos ver.
* **Se definen los rectángulos:** Se cargan tamaño de los rectángulos con los cuales interactuamos
* **Se definen las Fuentes:** Se definen las tipografías y el tamaño de los textos que usaremos.
* **Se definen las Imágenes:** Se definen una serie de imágenes que usaremos en el Juego
* **Se define la lista información:** Se definen una serie de diccionarios en los cuales se define la información por pregunta, la pregunta y su posición, las opciones y su posición, etc.
* **Se define las variables neutras:** Se definen una serie de valores usaremos para poder realizar el juego.
* **No llegamos a listar todo**

**Mencione las ventajas y desventajas de haber trabajado en “modo autosuficiente” (sin tanta asistencia del profe) y también la experiencia de haber trabajado en grupo de manera colaborativa:** Durante esta experiencia colaborativa pudimos obtener nuevo conocimiento sobre el tema orientado en la programación y pudimos desarrollar nuestras habilidades de manera autónoma, aunque con ayuda de distintas IA. Sin embargo, no todo fue positivo, ya que al empezar un tema desde cero tuvimos que buscar información por todas las fuentes posibles y aun así continuamos con muchas dudas, las cuales nos afectaron de manera negativa a la hora de programar y, en consecuencia, no llegamos a cumplir con los objetivos que nos habíamos propuesto al iniciar el segundo cuatrimestre.