



ESTRATEGIA DIDÁCTICA

Ayúdame, quiero ...

DATOS GENERALES

Nombre del participante	José Luis Sánchez López
Asignatura	Cibernética y Computación II
Año o semestre en que imparte	Sexto semestre
Horas clase a la semana	4 horas
Unidad	Unidad 2. Estructuras de control de secuencia en Java
Aprendizajes	<ul style="list-style-type: none">▪ Desarrolla programas que involucren las estructuras condicionales simples, compuestas y anidadas en los métodos de una Clase.▪ Desarrolla programas que involucren la estructura condicional múltiple en los métodos de una Clase.▪ Desarrolla programas para resolver problemas que involucren la estructura repetitiva for en los métodos de una Clase.▪ Desarrolla programas que involucren la estructura repetitiva while en los métodos de una Clase.▪ Realiza programas que involucren el uso de los arreglos bidimensionales▪ Desarrolla programas que involucren el uso de los arreglos bidimensionales en los métodos de una Clase▪ Desarrolla un proyecto que utilice las sentencias vistas hasta el momento, incluyendo los arreglos



Problemática que se abordará a través del problema.	<p>La enfermedad de Alzheimer se debe a cambios en el cerebro por la presencia de la proteína beta amiloide que se acumula frecuentemente en el lóbulo temporal. Dicha toxina provoca inflamación y muerte progresiva de neuronas.</p> <p>Provoca olvidos de eventos recientes, problemas de lenguaje, alteraciones del pensamiento abstracto, desorientación, olvido de lugares habituales, nombres de personas cercanas y vestirse por sí mismas, entre otras.</p>
Justificación.	<p>El programa generado en Python está orientado a apoyar al adulto mayor con demencia senil o la enfermedad de Alzheimer que presenta problemas para recordar el nombre de algunas actividades y expresarlas oralmente, mediante la presentación de imágenes que asocie con acciones cotidianas y la reproducción de audios que comuniquen el deseo del adulto mayor.</p> <p>Respecto a Cibernética y Computación II, la aplicación “Ayúdame, quiero ...” debe desarrollarse como un proyecto ya que permite al alumno poner en juego todos los aprendizajes adquiridos en la unidad 2, las estructuras de control de secuencia y que las aplique o implemente, las estructuras son las siguientes:</p> <ul style="list-style-type: none">▪ condicional simple, compuesta y anidada.▪ condicional múltiple.▪ de control repetitiva for▪ de control repetitiva while <p>Además de los arreglos bidimensionales y todas estas sentencias serán integradas en la elaboración del producto del proyecto, de forma que en clase se revise el concepto, la sentencia, se ejemplifique, se ejercite y después se implemente en el desarrollo de la aplicación.</p> <p>Acerca del lenguaje de programación, Java es un lenguaje totalmente orientado a objetos. Todas sus variables y funciones se definen dentro de clases. Por su parte, Python es un lenguaje multiparadigma, admite múltiples estilos de programación, incluidos el orientado a objetos, el procedimental y el funcional. Por lo tanto, Python es más flexible y compatible con muchas tareas.</p> <p>Java es un lenguaje compilado que se traduce a código máquina antes de ejecutarse. Mientras, Python es un lenguaje interpretado que se ejecuta en tiempo de ejecución. Esta diferencia afecta a la velocidad y eficacia de ambos lenguajes.</p>



	<p>Los programas Java pueden ejecutarse más rápido que los programas Python debido a su naturaleza compilada. Sin embargo, el método de interpretación interactiva puede hacer que los programas Python sean más fáciles de depurar y modificar.</p> <p>Debido a la flexibilidad, al método de interpretación interactiva, el tamaño de los programas que se desarrollan en la materia para lograr los aprendizajes es que me parece Python el mejor lenguaje.</p>
Producto esperado	<p>Una aplicación que utilice un adulto mayor, en ella identificará una acción que desee realizar mediante una imagen dentro de una pantalla y al presionarla se reproducirá un audio que exprese su intención, para que le ayuden a llevarla a cabo.</p> <p>Esta aplicación integrará quince imágenes que representaran ese número de acciones, así como quince audios uno para cada acción que se escucharán cuando se presione la imagen asociada.</p> <p>Para ello el alumno implementará al menos una vez, las sentencias condicional simple, compuesta y anidada, condicional múltiple, de control repetitiva for y while y un arreglo bidimensional.</p> <p>Las imágenes las obtendrá de repositorios de acceso libre, los audios, los generará empleando alguna aplicación en internet para producir audio a partir de texto, debe reportar tanto el o los repositorios y la aplicación para los audios.</p>
Recursos materiales /Herramientas TIC	<ul style="list-style-type: none">▪ Computadora o laptop.▪ Jupyter notebook o Visual studio code.▪ Lenguaje de programación Python.▪ Módulos pygame, sys y math.▪ Conexión a internet.▪ Plataforma educativa Moodle.▪ Videoprojector.▪ Pizarrón.▪ Plumón para pizarrón blanco.▪ Cronograma.▪ Listas de cotejo para evaluar el desarrollo del proyecto.
Tiempos de realización	10 horas una por clase y 2 extractase



Forma de trabajo	En equipos de forma colaborativa
------------------	----------------------------------

Secuencia didáctica



Presentación del problema a resolver

La enfermedad de Alzheimer se debe a cambios en el cerebro por la presencia de la proteína beta amiloide que se acumula frecuentemente en el lóbulo temporal. Dicha toxina provoca inflamación y muerte progresiva de neuronas.

Provoca olvidos de eventos recientes, problemas de lenguaje, alteraciones del pensamiento abstracto, desorientación, olvido de lugares habituales, nombres de personas cercanas y vestirse por sí mismas, entre otras.

Se requiere de un programa orientado a apoyar al adulto mayor con demencia senil o la enfermedad de Alzheimer que presenta problemas para recordar el nombre de algunas actividades y expresarlas oralmente, mediante la presentación de imágenes que asocie con acciones cotidianas y la reproducción de audios que comuniquen lo que quiere realizar el adulto mayor.



Inicio del proyecto

Actividad previa

- El profesor solicita la instalación en las computadoras del laboratorio de cómputo de los programas Visual Studio Code y Jupyter notebook

Sesión 1

- El profesor en la segunda hora de la primera sesión de la Unidad 2, presenta el proyecto, plantea el problema.

La enfermedad de Alzheimer se debe a cambios en el cerebro por la presencia de la proteína beta amiloide que se acumula frecuentemente en el lóbulo temporal. Dicha toxina provoca inflamación y muerte progresiva de neuronas.



Secuencia didáctica

Provoca olvidos de eventos recientes, problemas de lenguaje, alteraciones del pensamiento abstracto, desorientación, olvido de lugares habituales, nombres de personas cercanas y vestirse por sí mismas, entre otras.

El proyecto está destinado a crear una aplicación que utilice un adulto mayor, en ella identificara una acción que desee realizar mediante una imagen dentro de una pantalla y al presionarla se reproducirá un audio que exprese su intención, para que le ayuden a llevarla a cabo.

- El profesor organiza equipos de tres alumnos.
- El alumno anota los requerimientos del proyecto.



Desarrollo del proyecto

Sesión 2

- El profesor:
- Inicia un intercambio acerca de cómo abordarían la problemática, orientaría las participaciones a través del respeto y la tolerancia.
- explica que la aplicación:
- integrará quince imágenes que representaran ese número de acciones, así como quince audios uno para cada acción que se escucharán cuando se presione la imagen asociada.
- incluirá al menos una vez cada una de las sentencias condicional simple, compuesta y anidada, condicional múltiple, de control repetitiva for y while y un arreglo bidimensional.
- contendrá imágenes, estas las obtendrá de repositorios de acceso libre, los audios los generará empleando alguna aplicación en internet para producir audio a partir de texto. el alumno debe reportar tanto el o los repositorios y la aplicación para los audios.
- presenta los módulos requeridos

Sesión 3

- El profesor indica que activen en su computadora Jupyter notebook y su funcionamiento.
- El alumno entra Jupyter notebook, guiado por el profesor ubica los elementos del entorno de trabajo y ejercita los ejemplos que el profesor presenta.
- El profesor explica como importar los módulos, inicializar las funciones necesarias para la aplicación, definir el tamaño de la imagen y del botón para



Secuencia didáctica

iniciar la aplicación, así como definir colores que se utilizaran de fondo en la ventana o en los botones.

- El alumno:
- sigue las instrucciones y redacta tanto los comentarios que explican lo que se hará como las instrucciones que se ejecutarán.
- Da nombre al archivo, lo graba y lo guarda en la nube o en su memoria USB.

Sesión 4

- El profesor explica el funcionamiento de la aplicación empleando el mouse, luego señala puntualmente que acciones requieren programarse, luego encarga una investigación en equipo acerca de las instrucciones de pygame para el manejo de los eventos del mouse.
- El alumno en equipo indaga y elabora la respuesta a las preguntas de la investigación de forma colaborativa en un documento de texto compartido.
- El profesor lanzando pregunta a pregunta, en cada ocasión elige a un alumno diferente al azar de cada equipo.
- El alumno en turno contesta explicando la respuesta por equipo.
- El profesor retoma las sentencias condicionales y cuestiona a los equipos ¿Qué instrucciones utilizará para cerrar una ventana? y da tiempo de dos minutos
- Los equipos al terminar el tiempo dan respuesta.
- El profesor pregunta ¿Cómo será el código? y da cinco minutos
- Los equipos intercambian propuestas y uno de ellos expone su propuesta.
- El profesor aclara dudas, corrige y redacta el código.
- El alumno abre su archivo de trabajo en Jupyter notebook, transcribe el código y lo ejecuta.
- El profesor recuerda la sentencia switch y explica la sentencia para reproducir audio de pygame.
- Los equipos
- plantean una condicional compuesta, que emplee dos condiciones y para ejecutarse, se deben cumplir las dos condiciones.
- después redactan el código para que a partir de dos datos (número de caras y numero de aristas) presenten en pantalla el nombre del poliedro, los poliedros serán cubo, tetraedro, octaedro.
- depuran y corrige el código para que funcione de forma correcta.
- Guarda el archivo, al nombre le adiciona "v. 2.0", lo graba y lo guarda en la nube o en su memoria USB.

Sesión 5

- El profesor retoma la sentencia de control for y el manejo de eventos que hace pygame.



Secuencia didáctica

- El alumno abre su archivo de trabajo.
- El profesor guía el desarrollo del código del juego cho han, empleando el ciclo for para repetirlo determinado número de veces.
- El alumno
- aporta e implementa las sentencias condicionales simples, compuestas como repaso y el ciclo for para reafirmar la comprensión de su funcionamiento.
- depuran y corrige el código para que funcione de forma correcta.
- guarda el archivo, al nombre le adiciona "v. 3.0", lo graba y lo guarda en la nube o en su memoria USB.
- El profesor encarga quince imágenes y quince audios de acciones que los adultos mayores requieran ayuda para realizarlas. Las imágenes serán cuadradas con un tamaño de 200 pixeles en formato JPG. Los audios reproducirán en una frase la acción que ilustra la imagen correspondiente, tendrán una duración máxima de cinco segundos, serán grabados en formato MP3. Los traerán en una carpeta llamada recursos.

Sesión 6

- El profesor retoma la sentencia de control while y el manejo de eventos que hace pygame.
- El alumno abre su archivo de trabajo.
- El profesor guía el desarrollo del código del juego cho han, empleando el ciclo while para repetirlo hasta que ya no desee jugarlo.
- El alumno
- aporta e implementa las sentencias condicionales simples, compuestas como repaso y el ciclo while para reafirmar la comprensión de su funcionamiento.
- guarda el archivo, al nombre le adiciona "v. 4.0", lo graba y lo guarda en la nube o en su memoria USB.
- El profesor desarrollar el ciclo principal de la aplicación.
- Los equipos:
- siguen y aportan e implementan las sentencias condicionales y los ciclos repetitivos para crear el ciclo principal.
- depuran y corrigen el código para que funcione de forma correcta.
- guarda el archivo, al nombre le adiciona "v. 4.0", lo graba y lo guarda en la nube o en su memoria USB.

Sesión 7

- El profesor retoma el manejo de arreglos bidimensionales en Python.
- El alumno abre su archivo de trabajo.
- El profesor plantea un ejercicio de arreglo bidimensional.



Secuencia didáctica

- Los equipos:
- desarrollan el código para crear, almacenar los datos en el arreglo y mostrarlos en pantalla.
- definen la clase que presentara la tabla con las imágenes.
- crean la tabla de acciones insertando el nombre de cada una de las imágenes.
- definen el audio de cada acción.
- dimensionan la ventana de la aplicación.
- definen la fuente tipográfica para el botón de inicio.
- definen el tamaño y ubicación del botón de inicio.
- depuran y corrige el código para que funcione de forma correcta.
- guardan el archivo, al nombre le adiciona "v. 5.0", lo graba y lo guarda en la nube o en su memoria USB.

Sesión 8

- El profesor retoma.
- Los equipos:
- abren su archivo de trabajo.
- definen las funciones para iniciar y reiniciar las acciones.
- definen la ventana de la aplicación.
- dibujan el botón para iniciar la aplicación y ya dentro para seleccionar una acción.
- depuran y corrige el código para que funcione.
- ejecutan y prueban el funcionamiento de la aplicación
- guardan el archivo, al nombre le adiciona "v. 5.0", lo graba y lo guarda en la nube o en su memoria USB.
- Entregan su aplicación al profesor.



Cierre del proyecto

Los equipos muestran sus aplicaciones

Un alumno elegido por el profesor recapitula los temas que formaron parte del proyecto.

El profesor promueve que los equipos formulen conclusiones.



Secuencia didáctica



Evaluación

ACTIVIDAD	VALOR
Ejecución, trabajo colaborativo participación e importancia de las aportaciones, la resolución de conflictos y la cohesión del grupo.	3
Ejecución, trabajó con respeto y tolerancia.	3
Producto final, la aplicación se generó integrando las estructuras de control de secuencia indicadas en el proyecto.	3
Producto final, la aplicación realiza la operación para lo que se diseñó y es funcional.	1



Referencias

INTEF. (1 de diciembre de 2021,). *PyGame: Realizando juegos con Python* - Code INTEF. Recuperado 3 de julio de 2023, de https://code.intef.es/prop_didacticas/pygame-realizando-juegos-con-python/

PyGame v2.5.0 documentation. (s.f.). Recuperado 24 de junio de 2023, de <https://www.pygame.org/docs/index.html>

Tagliaferri, L. (2018). *How To Code in Python 3* [PDF]. Estados Unidos: DigitalOcean. Recuperado de <https://assets.digitalocean.com/books/python/how-to-code-in-python.pdf>



Anexo. Código de la aplicación Ayúdame, quiero ...

```
# Importación de los módulos requeridos
import pygame
import sys
import math

# Iniciación de las funciones de Pygame para usar la tipografía y el sonido
pygame.init()
pygame.font.init()
pygame.mixer.init()

# Tamaño del dibujo y del botón de inicio
medida_dibujo = 200 # tamaño de las imágenes en pixeles
alto_boton = 30 # El botón inferior, para iniciar la aplicación

# Definición de colores
azul = (30, 136, 229)
gris = (206, 206, 206)
blanco = (255, 255, 255)
negro = (0, 0, 0)

# Creación de la clase que presentara la tabla con las imágenes.
class Tabla:
    def __init__(self, fuente_imagen):
        self.mostrar = True
        self.descubierto = False
        self.fuente_imagen = fuente_imagen
        self.imagen_real = pygame.image.load(fuente_imagen)

# Creación de la tabla de acciones
acciones = [
    [Tabla("recursos/Baniar.png"), Tabla("recursos/Caer.png"),
     Tabla("recursos/Caminar.jpg"), Tabla("recursos/Cenar.png"),
     Tabla("recursos/Comer.jpg")],
    [Tabla("recursos/Desayunar.png"), Tabla("recursos/Dormir.jpg"),
     Tabla("recursos/Ejercicio.png"), Tabla("recursos/Frio.png"),
     Tabla("recursos/lavar_dientes.png")],
    [Tabla("recursos/lavar_manos.png"), Tabla("recursos/Leer.png"),
     Tabla("recursos/Pintar.png"), Tabla("recursos/Pipipopo.png"),
     Tabla("recursos/Ver_TV.jpg")],
]

# Declaración del audio de cada acción
sonido_clic = pygame.mixer.Sound("recursos/clic.wav")
sonido_baniar = pygame.mixer.Sound("recursos/baniar.mp3")
sonido_caer = pygame.mixer.Sound("recursos/caer.mp3")
sonido_caminar = pygame.mixer.Sound("recursos/caminar.mp3")
sonido_cenar = pygame.mixer.Sound("recursos/cenar.mp3")
sonido_comer = pygame.mixer.Sound("recursos/comer.mp3")
sonido_desayunar = pygame.mixer.Sound("recursos/desayunar.mp3")
sonido_dormir = pygame.mixer.Sound("recursos/dormir.mp3")
sonido_ejercicio = pygame.mixer.Sound("recursos/ejercicio.mp3")
sonido_frio = pygame.mixer.Sound("recursos/frio.mp3")
sonido_lavar_dientes = pygame.mixer.Sound("recursos/lavar_dientes.mp3")
sonido_lavar_manos = pygame.mixer.Sound("recursos/lavar_manos.mp3")
sonido_leer = pygame.mixer.Sound("recursos/leer.mp3")
sonido_pintar = pygame.mixer.Sound("recursos/pintar.mp3")
sonido_pipipopo = pygame.mixer.Sound("recursos/pipipopo.mp3")
sonido_ver_tv = pygame.mixer.Sound("recursos/ver_tv.mp3")
sonido_terminar = pygame.mixer.Sound("recursos/terminar.mp3")
```



```
# Dimensionamiento de la pantalla a partir del tamaño de los dibujos
ancho_pantalla = len(acciones[0]) * medida_dibujo
alto_pantalla = (len(acciones) * medida_dibujo) + alto_boton
ancho_boton = ancho_pantalla

# Definición de la fuente para el botón
tamano_fuente = 20
fuente = pygame.font.SysFont("Arial", tamano_fuente)
xFuente = int((ancho_boton / 2) - (tamano_fuente / 2))
yFuente = int(alto_pantalla - alto_boton)

# Definición del botón rectangular inferior
boton = pygame.Rect(0, alto_pantalla - alto_boton, ancho_boton, alto_pantalla)

# Bandera para saber si reaccionar a los eventos del usuario
puede_iniciar = True

# Saber si la aplicación está iniciada
aplicacion_iniciada = False

# Funciones empleadas
def reiniciar_acciones():
    global aplicacion_iniciada
    aplicacion_iniciada = False

def iniciar_acciones():
    pygame.mixer.Sound.play(sonido_clic)
    global aplicacion_iniciada
    aplicacion_iniciada = True

# Definición de la pantalla con las medidas previamente calculadas y se coloca el
título de la ventana
pantalla = pygame.display.set_mode((ancho_pantalla, alto_pantalla))
pygame.display.set_caption('Ayúdame, quiero ...')

# Ciclo en el que se recibe la elección de la acción y la reproducción de su audio
while True:
    # Escuchar eventos, pues estamos en un ciclo infinito que se repite varias veces
    por segundo
    for event in pygame.event.get():
        if event.type == pygame.QUIT:      # Si presiona Cerrar, salimos de la aplicación
            sys.exit()
        elif event.type == pygame.MOUSEBUTTONDOWN and puede_iniciar: # Se hizo clic en
una acción
            xAbsoluto, yAbsoluto = event.pos # Coordenadas en donde se hizo clic.
            if boton.collidepoint(event.pos): # Si se hizo click en el botón y la
aplicación no se ha iniciado, la iniciamos
                if not aplicacion_iniciada:
                    iniciar_acciones()
            else:
                if not aplicacion_iniciada: # Si la aplicación no se ha iniciado,
ignoramos el clic
                    continue

            # Determinación de X e Y, siendo estos los índices de la tabla
            x = math.floor(xAbsoluto / medida_dibujo)
            y = math.floor(yAbsoluto / medida_dibujo)

            # Ubica la acción en la tabla con las coordenadas x,y
            Tabla = acciones[y][x]

            # Reproduducir el audio correspondiente a partir de las coordenadas x,y
```



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

COLEGIO DE CIENCIAS Y HUMANIDADES



```
if y == 0 and x == 0:
    pygame.mixer.Sound.play(sonido_banar)
    pygame.display.set_caption('bañar')
elif y == 0 and x == 1:
    pygame.mixer.Sound.play(sonido_caer)
elif y == 0 and x == 2:
    pygame.mixer.Sound.play(sonido_caminar)
elif y == 0 and x == 3:
    pygame.mixer.Sound.play(sonido_cenar)
elif y == 0 and x == 4:
    pygame.mixer.Sound.play(sonido_comer)
elif y == 1 and x == 0:
    pygame.mixer.Sound.play(sonido_desayunar)
elif y == 1 and x == 1:
    pygame.mixer.Sound.play(sonido_dormir)
elif y == 1 and x == 2:
    pygame.mixer.Sound.play(sonido_ejercicio)
elif y == 1 and x == 3:
    pygame.mixer.Sound.play(sonido_frio)
elif y == 1 and x == 4:
    pygame.mixer.Sound.play(sonido_lavar_dientes)
elif y == 2 and x == 0:
    pygame.mixer.Sound.play(sonido_lavar_manos)
elif y == 2 and x == 1:
    pygame.mixer.Sound.play(sonido_leer)
elif y == 2 and x == 2:
    pygame.mixer.Sound.play(sonido_pintar)
elif y == 2 and x == 3:
    pygame.mixer.Sound.play(sonido_pipipopo)
elif y == 2 and x == 4:
    pygame.mixer.Sound.play(sonido_ver_tv)

# Rellenar el fondo de la pantalla en blanco
pantalla.fill( blanco )

# Banderas para saber en dónde dibujar las imágenes
x = 0
y = 0

# Recorrer las acciones
for fila in acciones:
    x = 0
    for Tabla in fila:
        # Dibuja la imagen
        pantalla.blit( Tabla.imagen_real, (x, y) )
        x += medida_dibujo
    y += medida_dibujo

# Dibuja el botón para iniciar la aplicación y ya dentro para seleccionar una acción
if aplicacion_iniciada:
    pygame.draw.rect( pantalla, negro, boton )
    pantalla.blit( fuente.render(
        "Selecciona una acción", True, blanco ), (xFuente, yFuente) )
else:
    pygame.draw.rect( pantalla, azul, boton )
    pantalla.blit( fuente.render(
        "Presiona aquí para iniciarla aplicación", True, negro ), (xFuente,
yFuente) )

# Actualizamos la pantalla
pygame.display.update()
```