

Universidad Americana



Algoritmos y Estructuras de Datos

Integrante:

Gabriel Antonio Rojas Uriarte

Docente:

Cesar Marín

Mayo 12 de 2025

Ejercicio 1:

```
class ColaImpresion:
    def __init__(self):
        self.documentos = []

    #Enqueue: Agregar documento a la cola
    def addDocumento(self, nombre, usuario, paginas):
        documento = {
            "nombre": nombre,
            "usuario": usuario,
            "paginas": paginas
        }
        self.documentos.append(documento)
        print(f"Se ha añadido el documento {nombre} de {usuario} con {paginas} páginas a la cola")

    #Dequeue: Eliminar el primer documento de la cola (o el mas antiguo)
    def procesarDocumento(self):
        if self.documentos:
            documento = self.documentos.pop(0)
            print("Se ha procesado el documento")
        else:
            print("No hay documentos en la cola")

    def imprimirDocumento(self):
        if self.documentos:
            documento = self.documentos[0]
            print(f"Se está imprimiendo el documento {documento['nombre']} de {documento['usuario']}")
        else:
            print("No hay documentos en la cola")

    #Verificar si la cola está vacía
    def vacia(self):
        return len(self.documentos) == 0

def menu():
    cola = ColaImpresion()
    while True:
        print("\nCola de impresión de documentos")
        print("1. Agregar documento a la cola")
        print("2. Procesar documento")
        print("3. Mostrar documento en proceso")
        print("4. Salir")
        opcion = int(input("Seleccione una opcion: "))
        if opcion == 1:
            nombre = input("Ingrese el nombre del documento: ")
            usuario = input("Ingrese el nombre del usuario que lo envió: ")
            paginas = int(input("Ingrese el número de páginas: "))
            cola.addDocumento(nombre, usuario, paginas)
        elif opcion == 2:
            cola.procesarDocumento()
        elif opcion == 3:
            cola.imprimirDocumento()
        elif opcion == 4:
            print("Saliendo del programa")
            break
        else:
            print("Opcion invalida, intentelo nuevamente")

if __name__ == "__main__":
    menu()
```

Terminal:

```
Cola de impresión de documentos
1. Agregar documento a la cola
2. Procesar documento
3. Mostrar documento en proceso
4. Salir
Seleccione una opcion: 1
Ingrese el nombre del documento: Ejercicios de python
Ingrese el nombre del usuario que lo envió: Gabriel
Ingrese el número de páginas: 2
Se ha añadido el documento Ejercicios de python de Gabriel con 2 páginas a la cola
```

```
Cola de impresión de documentos
1. Agregar documento a la cola
2. Procesar documento
3. Mostrar documento en proceso
4. Salir
Seleccione una opcion: 1
Ingrese el nombre del documento: blog de notas
Ingrese el nombre del usuario que lo envió: Gabo
Ingrese el número de páginas: 3
Se ha añadido el documento blog de notas de Gabo con 3 páginas a la cola
```

```
Cola de impresión de documentos
1. Agregar documento a la cola
2. Procesar documento
3. Mostrar documento en proceso
4. Salir
Seleccione una opcion: 3
Se está imprimiendo el documento Ejercicios de python de Gabriel)
```

```
Cola de impresión de documentos
1. Agregar documento a la cola
2. Procesar documento
3. Mostrar documento en proceso
4. Salir
Seleccione una opcion: 2
Se ha procesado el documento
```

```
Cola de impresión de documentos
1. Agregar documento a la cola
2. Procesar documento
3. Mostrar documento en proceso
4. Salir
Seleccione una opcion: 3
Se está imprimiendo el documento blog de notas de Gabo)
```

```
Cola de impresión de documentos
1. Agregar documento a la cola
2. Procesar documento
3. Mostrar documento en proceso
4. Salir
Seleccione una opcion: 2
Se ha procesado el documento
```

```
Cola de impresión de documentos
1. Agregar documento a la cola
2. Procesar documento
3. Mostrar documento en proceso
4. Salir
Seleccione una opcion: 4
Saliendo del programa
```

```
PS C:\Users\gabri\OneDrive\Documentos\Algoritmos\Semana8\Guia didactica - Ejercicios de cola> █
```

Ejercicio 2:

```
class Cola:
    def __init__(self):
        self.llamadas = []

    def agregar(self, nombre, motivo):
        llamada = {
            "nombre": nombre,
            "motivo": motivo,
        }
        self.llamadas.append(llamada)
        print(f"Se ha añadido la llamada {nombre} con motivo {motivo} a la cola")

    def atender(self):
        if self.llamadas:
            llamada = self.llamadas.pop(0)
            print(f"Se ha atendido la llamada {llamada['nombre']} con motivo {llamada['motivo']}")
        else:
            print("No hay llamadas en la cola")

    def imprimir(self):
        if self.llamadas:
            print("Lista de llamadas en espera de atención: ")
            for i, llamada in enumerate(self.llamadas, start=1):
                print(f"{i}. {llamada['nombre']} con motivo {llamada['motivo']}")
        else:
            print("No hay llamadas en espera")

def menu():
    cola = Cola()
    while True:
        print("\nPrograma de atención de llamadas")
        print("1. Agregar llamada a la cola")
        print("2. Atender llamada")
        print("3. Imprimir lista de llamadas")
        print("4. Salir")
        opcion = int(input("Seleccione una opcion: "))
        if opcion == 1:
            nombre = input("Ingrese el nombre del cliente: ")
            motivo = input("Ingrese el motivo de la llamada: ")
            cola.agregar(nombre, motivo)
        elif opcion == 2:
            cola.atender()
            print("llamada atendida")
        elif opcion == 3:
            cola.imprimir()
        elif opcion == 4:
            print("Saliendo del programa")
            break
        else:
            print("Opción invalida, intentelo nuevamente")

if __name__ == "__main__":
    menu()
```

Terminal:

```
Programa de atención de llamadas
1. Agregar llamada a la cola
2. Atender llamada
3. Imprimir lista de llamadas
4. Salir
Seleccione una opcion: 1
Ingrese el nombre del cliente: Gabriel
Ingrese el motivo de la llamada: Urgencia
Se ha añadido la llamada Gabriel con motivo Urgencia a la cola

Programa de atención de llamadas
1. Agregar llamada a la cola
2. Atender llamada
3. Imprimir lista de llamadas
4. Salir
Seleccione una opcion: 1
Ingrese el nombre del cliente: Cliente1
Ingrese el motivo de la llamada: Enfermedad
Se ha añadido la llamada Cliente1 con motivo Enfermedad a la cola

Programa de atención de llamadas
1. Agregar llamada a la cola
2. Atender llamada
3. Imprimir lista de llamadas
4. Salir
Seleccione una opcion: 3
Lista de llamadas en espera de atención:
1. Gabriel con motivo Urgencia
2. Cliente1 con motivo Enfermedad

Programa de atención de llamadas
1. Agregar llamada a la cola
2. Atender llamada
3. Imprimir lista de llamadas
4. Salir
Seleccione una opcion: 2
Se ha atendido la llamada Gabriel con motivo Urgencia
llamada atendida

Programa de atención de llamadas
1. Agregar llamada a la cola
2. Atender llamada
3. Imprimir lista de llamadas
4. Salir
Seleccione una opcion: 3
Lista de llamadas en espera de atención:
1. Cliente1 con motivo Enfermedad

Programa de atención de llamadas
1. Agregar llamada a la cola
2. Atender llamada
3. Imprimir lista de llamadas
4. Salir
Seleccione una opcion: 2
Se ha atendido la llamada Cliente1 con motivo Enfermedad
llamada atendida

Programa de atención de llamadas
1. Agregar llamada a la cola
2. Atender llamada
3. Imprimir lista de llamadas
4. Salir
Seleccione una opcion: 3
No hay llamadas en espera
```

Programa de atención de llamadas

1. Agregar llamada a la cola

2. Atender llamada

3. Imprimir lista de llamadas

4. Salir

Seleccione una opcion: 4

Saliendo del programa

PS C:\Users\gabri\OneDrive\Documentos\Algoritmos\Semana8\Guia didactica - Ejercicios de cola> █

Ejercicio 3:

```
class Cola:
    def __init__(self):
        self.Pacientes = []

    def agregar(self, nombre, tipo):
        paciente = {
            "nombre": nombre,
            "tipo": tipo,
            "turno": None,
        }
        self.Pacientes.append(paciente)
        print(f"Se ha añadido el paciente {nombre} con tipo {tipo} a la cola")

    def atender(self):
        if self.Pacientes:
            paciente = self.Pacientes.pop(0)
            print(f"Se ha atendido el paciente {paciente['nombre']} con tipo {paciente['tipo']}")
        else:
            print("No hay pacientes en la cola")

    def imprimir(self):
        if self.Pacientes:
            print("Lista de pacientes en espera de atención: ")
            for i, paciente in enumerate(self.Pacientes, start=1):
                print(f"{i}. {paciente['nombre']} con tipo {paciente['tipo']}")
        else:
            print("No hay pacientes en espera")

def main():
    cola = Cola()
    while True:
        print("\nPrograma de atención de pacientes")
        print("1. Agregar paciente a la cola")
        print("2. Atender paciente")
        print("3. Imprimir lista de pacientes")
        print("4. Salir")
        opcion = int(input("Seleccione una opcion: "))
        if opcion == 1:
            nombre = input("Ingrese el nombre del paciente: ")
            tipo = input("Ingrese el tipo de servicio: ")
            cola.agregar(nombre, tipo)
        elif opcion == 2:
            cola.atender()
            print("Paciente atendido")
        elif opcion == 3:
            cola.imprimir()
        elif opcion == 4:
            print("Saliendo del programa")
            break
        else:
            print("Opción invalida, intentelo nuevamente")

if __name__ == "__main__":
    main()
```

Terminal:

```
Programa de atención de pacientes
1. Agregar paciente a la cola
2. Atender paciente
3. Imprimir lista de pacientes
4. Salir
Seleccione una opcion: 1
Ingrese el nombre del paciente: Gabriel
Ingrese el tipo de servicio: Limpieza de cuarto
Se ha añadido el paciente Gabriel con tipo Limpieza de cuarto a la cola

Programa de atención de pacientes
1. Agregar paciente a la cola
2. Atender paciente
3. Imprimir lista de pacientes
4. Salir
Seleccione una opcion: 1
Ingrese el nombre del paciente: Johanneris
Ingrese el tipo de servicio: Emergencia
Se ha añadido el paciente Johanneris con tipo Emergencia a la cola

Programa de atención de pacientes
1. Agregar paciente a la cola
2. Atender paciente
3. Imprimir lista de pacientes
4. Salir
Seleccione una opcion: 3
Lista de pacientes en espera de atención:
1. Gabriel con tipo Limpieza de cuarto
2. Johanneris con tipo Emergencia
No hay pacientes en espera

Programa de atención de pacientes
1. Agregar paciente a la cola
2. Atender paciente
3. Imprimir lista de pacientes
4. Salir
Seleccione una opcion: 2
Se ha atendido el paciente Gabriel con tipo Limpieza de cuarto
Paciente atendido

Programa de atención de pacientes
1. Agregar paciente a la cola
2. Atender paciente
3. Imprimir lista de pacientes
4. Salir
Seleccione una opcion: 3
Lista de pacientes en espera de atención:
1. Johanneris con tipo Emergencia
No hay pacientes en espera

Programa de atención de pacientes
1. Agregar paciente a la cola
2. Atender paciente
3. Imprimir lista de pacientes
4. Salir
Seleccione una opcion: 2
Se ha atendido el paciente Johanneris con tipo Emergencia
Paciente atendido
```


Programa de atención de pacientes

1. Agregar paciente a la cola
2. Atender paciente
3. Imprimir lista de pacientes
4. Salir

Seleccione una opcion: 3

Programa de atención de pacientes

1. Agregar paciente a la cola
2. Atender paciente
3. Imprimir lista de pacientes
4. Salir

Seleccione una opcion: 4

Saliendo del programa

PS C:\Users\gabri\OneDrive\Documentos\Algoritmos\Semana8\Guia didactica - Ejercicios de cola> █

Ejercicio 4:

```
class Proceso:
    def __init__(self):
        self.proceso = []

    def agregar(self, id, nombre, duracion):
        proceso = {
            "id": id,
            "nombre": nombre,
            "duración": duracion,
        }
        self.proceso.append(proceso)
        print(f"Se ha añadido el proceso {id} con nombre {nombre} y duración {duracion} a la cola")

    def procesar(self):
        if self.proceso:
            proceso = self.proceso.pop(0)
            print("Se ha procesado exitosamente")
        else:
            print("No hay procesos en la cola")

    def imprimir(self):
        if self.proceso:
            proceso = self.proceso[0]
            print(f"Se esta procesando el proceso {proceso['id']} con nombre {proceso['nombre']}")
        else:
            print("No hay procesos en la cola")

    def vacia(self):
        return len(self.proceso) == 0

def main():
    proceso = Proceso()
    while True:
        print("\nMicroprocesador de procesos")
        print("1. Agregar proceso a la cola")
        print("2. Procesar")
        print("3. Mostrar proceso en ejecución")
        print("4. Salir")
        opcion = int(input("Seleccione una opcion: "))
        if opcion == 1:
            id = int(input("Ingrese el id del proceso: "))
            nombre = input("Ingrese el nombre del proceso: ")
            duracion = int(input("Ingrese la duración estimada del proceso en milisegundos: "))
            proceso.agregar(id, nombre, duracion)
        elif opcion == 2:
            proceso.procesar()
        elif opcion == 3:
            proceso.imprimir()
        elif opcion == 4:
            print("Saliendo del programa")
            break
        else:
            print("Opción invalida, intentelo nuevamente")

if __name__ == "__main__":
    main()
```

Terminal:

```
Microprocesador de procesos
1. Agregar proceso a la cola
2. Procesar
3. Mostrar proceso en ejecución
4. Salir
Seleccione una opcion: 1
Ingrese el id del proceso: 1
Ingrese el nombre del proceso: Optimizacion
Ingrese la duración estimada del proceso en milisegundos: 1
Se ha añadido el proceso 1 con nombre Optimizacion y duración 1 a la cola

Microprocesador de procesos
1. Agregar proceso a la cola
2. Procesar
3. Mostrar proceso en ejecución
4. Salir
Seleccione una opcion: 3
Se esta procesando el proceso 1 con nombre Optimizacion

Microprocesador de procesos
1. Agregar proceso a la cola
2. Procesar
3. Mostrar proceso en ejecución
4. Salir
Seleccione una opcion: 2
Se ha procesado exitosamente

Microprocesador de procesos
1. Agregar proceso a la cola
2. Procesar
3. Mostrar proceso en ejecución
4. Salir
Seleccione una opcion: 4
Saliendo del programa
PS C:\Users\gabri\OneDrive\Documentos\Algoritmos\Semana8\Guia didactica - Ejercicios de cola> |
```

Ejercicio 5:

```
class Servidor:
    def __init__(self):
        self.solicitudes = []

    def agregar(self, nombre, archivo):
        solicitud = {
            "nombre": nombre,
            "archivo": archivo,
        }
        self.solicitudes.append(solicitud)
        print(f"Se ha añadido la solicitud de {nombre} para el archivo {archivo} a la cola")

    def procesar(self):
        if self.solicitudes:
            solicitud = self.solicitudes.pop(0)
            print(f"Se ha procesado la solicitud de {solicitud['nombre']} para el archivo {solicitud['archivo']}")
        else:
            print("No hay solicitudes pendientes")

    def imprimir(self):
        if self.solicitudes:
            print("Lista de solicitudes pendientes: ")
            for i, solicitud in enumerate(self.solicitudes, start=1):
                print(f"{i}. Solicitud de {solicitud['nombre']} para el archivo {solicitud['archivo']}")
        else:
            print("No hay solicitudes pendientes")

    def vacia(self):
        return len(self.solicitudes) == 0

def main():
    server = Servidor()
    while True:
        print("\nPrograma de atención de solicitudes")
        print("1. Agregar solicitud a la cola")
        print("3. Mostrar solicitud pendiente")
        print("4. Salir")
        opcion = int(input("Seleccione una opcion: "))
        if opcion == 1:
            nombre = input("Ingrese el nombre del usuario: ")
            archivo = input("Ingrese el nombre del archivo: ")
            server.agregar(nombre, archivo)
        elif opcion == 2:
            server.procesar()
        elif opcion == 3:
            server.imprimir()
        elif opcion == 4:
            print("Saliendo del programa")
            break
        else:
            print("Opción invalida, intentelo nuevamente")

if __name__ == "__main__":
    main()
```

Terminal:

```
Programa de atención de solicitudes
1. Agregar solicitud a la cola
3. Mostrar solicitud pendiente
4. Salir
Seleccione una opcion: 1
Ingrese el nombre del usuario: Gabriel
Ingrese el nombre del archivo: Ejercicio
Se ha añadido la solicitud de Gabriel para el archivo Ejercicio a la cola

Programa de atención de solicitudes
1. Agregar solicitud a la cola
3. Mostrar solicitud pendiente
4. Salir
Seleccione una opcion: 1
Ingrese el nombre del usuario: Johaneris
Ingrese el nombre del archivo: Compilador
Se ha añadido la solicitud de Johaneris para el archivo Compilador a la cola

Programa de atención de solicitudes
1. Agregar solicitud a la cola
3. Mostrar solicitud pendiente
4. Salir
Seleccione una opción: 3
Lista de solicitudes pendientes:
1. Solicitud de Gabriel para el archivo Ejercicio
2. Solicitud de Johaneris para el archivo Compilador

Programa de atención de solicitudes
1. Agregar solicitud a la cola
3. Mostrar solicitud pendiente
4. Salir
Seleccione una opcion: 4
Saliendo del programa
PS C:\Users\gabri\OneDrive\Documentos\Algoritmos\Semana8\Guia didactica - Ejercicios de cola>
```