

UNIVERSIDAD CONTINENTAL

# Finproc

Docente: Yesenia Concha  
Ramos

Integrantes:

- Ccarita Velasquez , Jhon  
Gustavo
- Sevillanos Tinco , Rodrigo  
Antonio
- Espinoza Zea , Andre  
Sebastián

NRC: 59098

Cusco- Perú  
2025-02



# Planteamiento del problema

En las entidades financieras, la atención al cliente suele presentar demoras y desorganización debido a la concurrencia de múltiples operaciones como depósitos, retiros o solicitudes de préstamos. Esta situación genera largos tiempos de espera, errores en el registro de transacciones y falta de control en el flujo de atención.

El objetivo de este proyecto es desarrollar un sistema de simulación que organice y gestione de manera eficiente el proceso de atención bancaria, aplicando estructuras de datos dinámicas lineales (listas enlazadas, colas y pilas). Con ello se busca optimizar el orden, reducir los tiempos de espera y mejorar el registro y trazabilidad de las operaciones.

# Descripción de estructuras de datos y operaciones

## Lista Enlazada – Gestión de Clientes

```
struct Cliente {  
    char dni[9];           // Identificador único del cliente (8 dígitos)  
    char nombre[50];       // Nombre completo del cliente  
    char tipo[15];         // Tipo de cliente: VIP, Preferencial o Regular  
    Cliente *sig;          // Puntero al siguiente nodo en la lista  
};|
```

# Descripción de estructuras de datos y operaciones

## Cola de Prioridad – Gestión de Atención

```
struct NodoCola {  
    char dni[9];           // DNI del cliente en cola  
    int prioridad;         // Nivel de prioridad (1 = VIP, 2 = Preferencial, 3 = Regular)  
    NodoCola *sig;        // Puntero al siguiente nodo en la cola  
};
```

## Pila – Registro de Transacciones

```
struct Transaccion {  
    char tipo[20];         // Tipo de transacción (depósito/retiro)  
    double monto;          // Monto de la operación  
    Transaccion *sig;      // Enlace al siguiente nodo (anterior transacción)  
};|
```

# Algoritmos principales: Pseudocódigo para agregar proceso.

Pseudocódigo para  
agregar proceso.

```
Algoritmo Agregar_Proceso
Entrada: Datos del proceso (DNI, nombre, tipo o transacción)
Salida: Estructura actualizada con el nuevo nodo insertado

1. Crear un nuevo nodo N
2. Leer los datos de entrada y asignarlos a N
3. Si se trata de un cliente:
    a. Si la lista está vacía:
        lista ← N
    b. En caso contrario:
        recorrer la lista hasta el último nodo
        enlazar el último nodo con N
    FinSi
4. Si se trata de una transacción:
    a. enlazar N con el nodo que ocupa el tope de la pila
    b. actualizar el puntero del tope → N
    FinSi
5. Mostrar mensaje de confirmación
6. Fin del algoritmo
```

# Algoritmos principales: Pseudocódigo para agregar proceso.

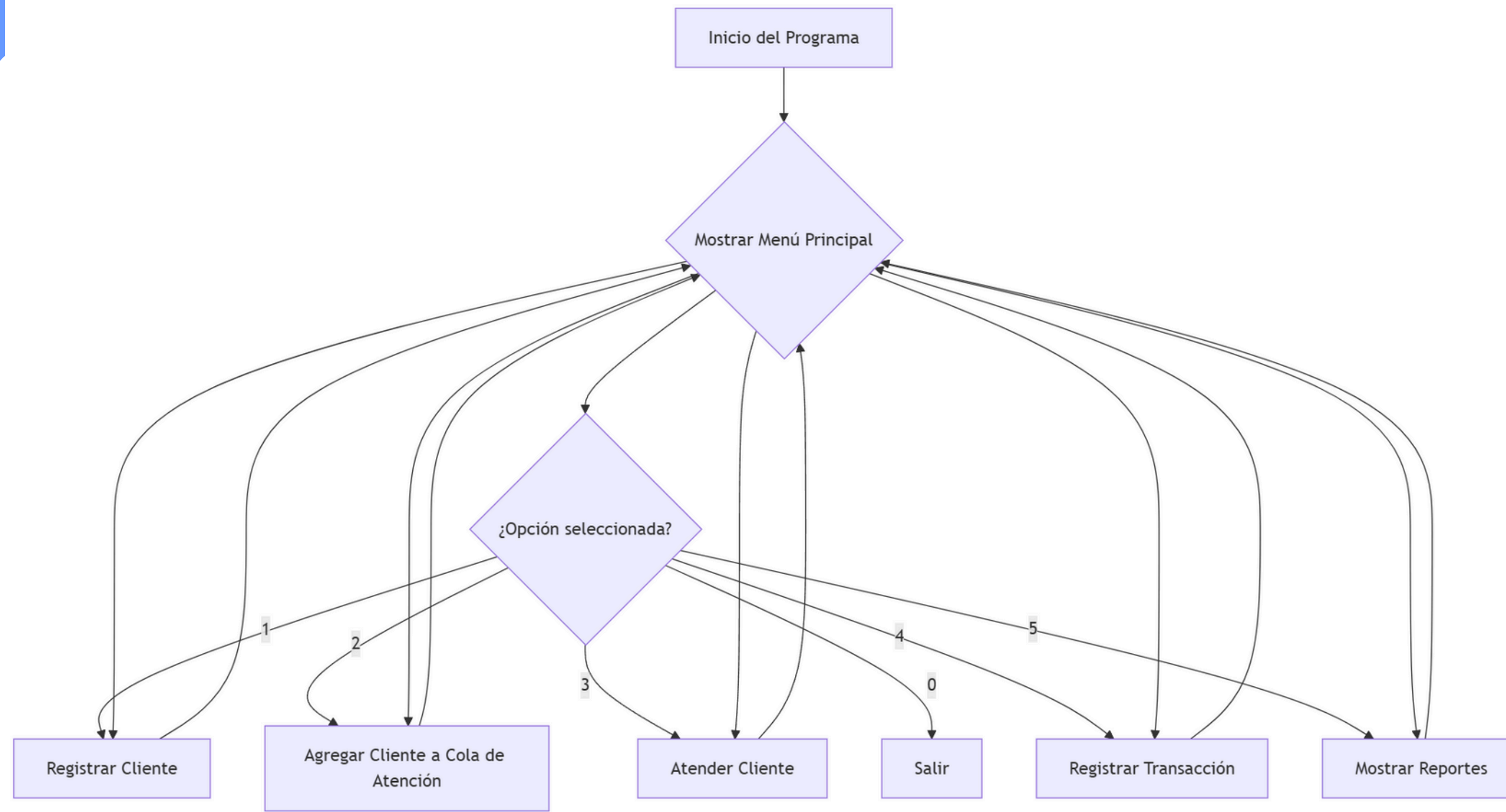
Pseudocódigo para  
cambiar estado del  
proceso.

```
Algoritmo Cambiar_Estado_Proceso
Entrada: Cola de prioridad con clientes en espera
Salida: Cliente atendido y cola actualizada

1. Si la cola está vacía:
    Mostrar "No hay clientes en cola"
    Terminar algoritmo
2. FinSi
3. Mostrar datos del primer cliente (DNI, tipo)
4. Eliminar el primer nodo de la cola (cliente atendido)
5. Actualizar el puntero principal de la cola → siguiente nodo
6. Liberar memoria del nodo eliminado
7. Mostrar mensaje "Cliente atendido con éxito"
8. Fin del algoritmo
```

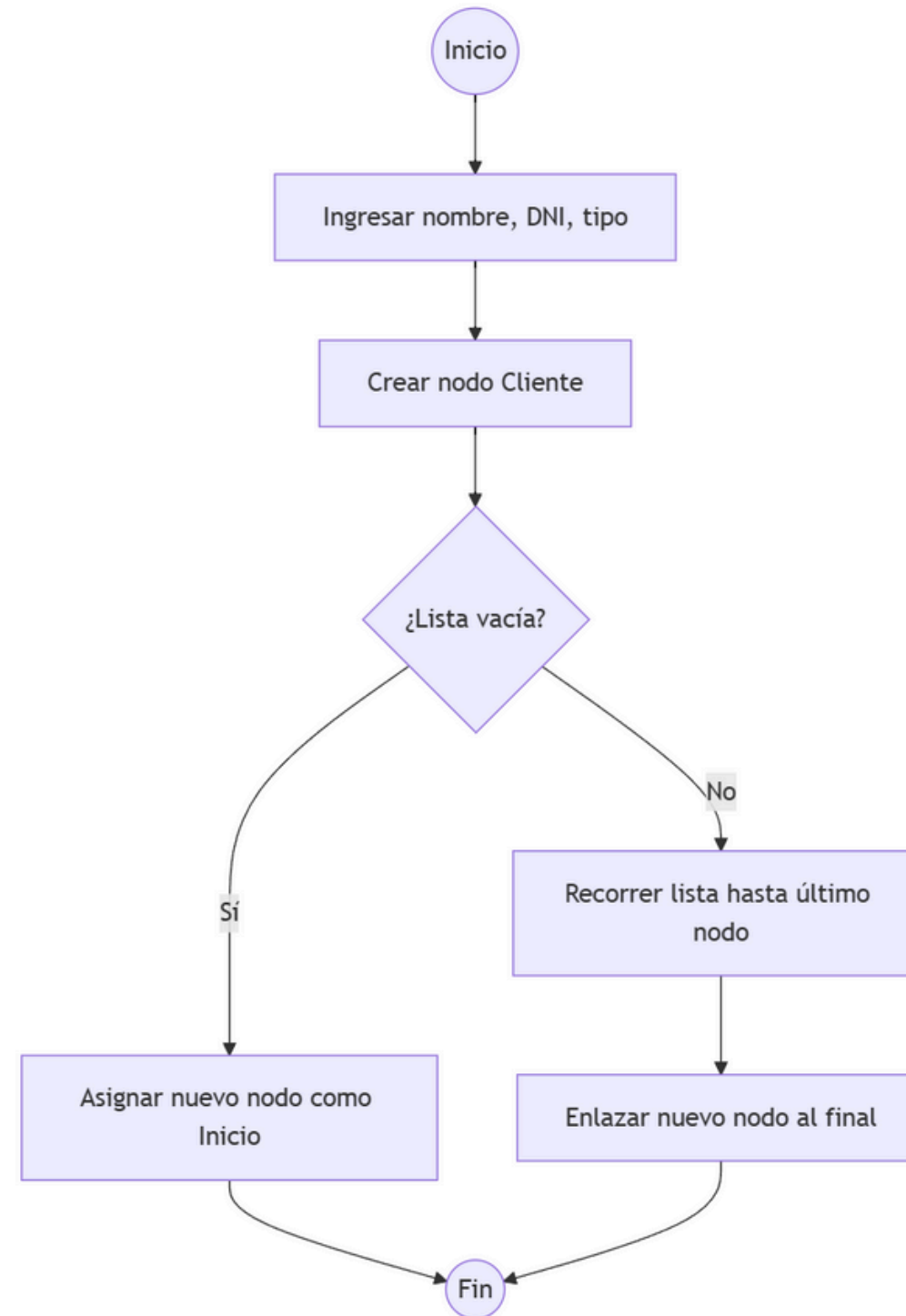
# Diagramas de Flujo

## Diagrama de flujo principal:



# Diagramas de Flujo

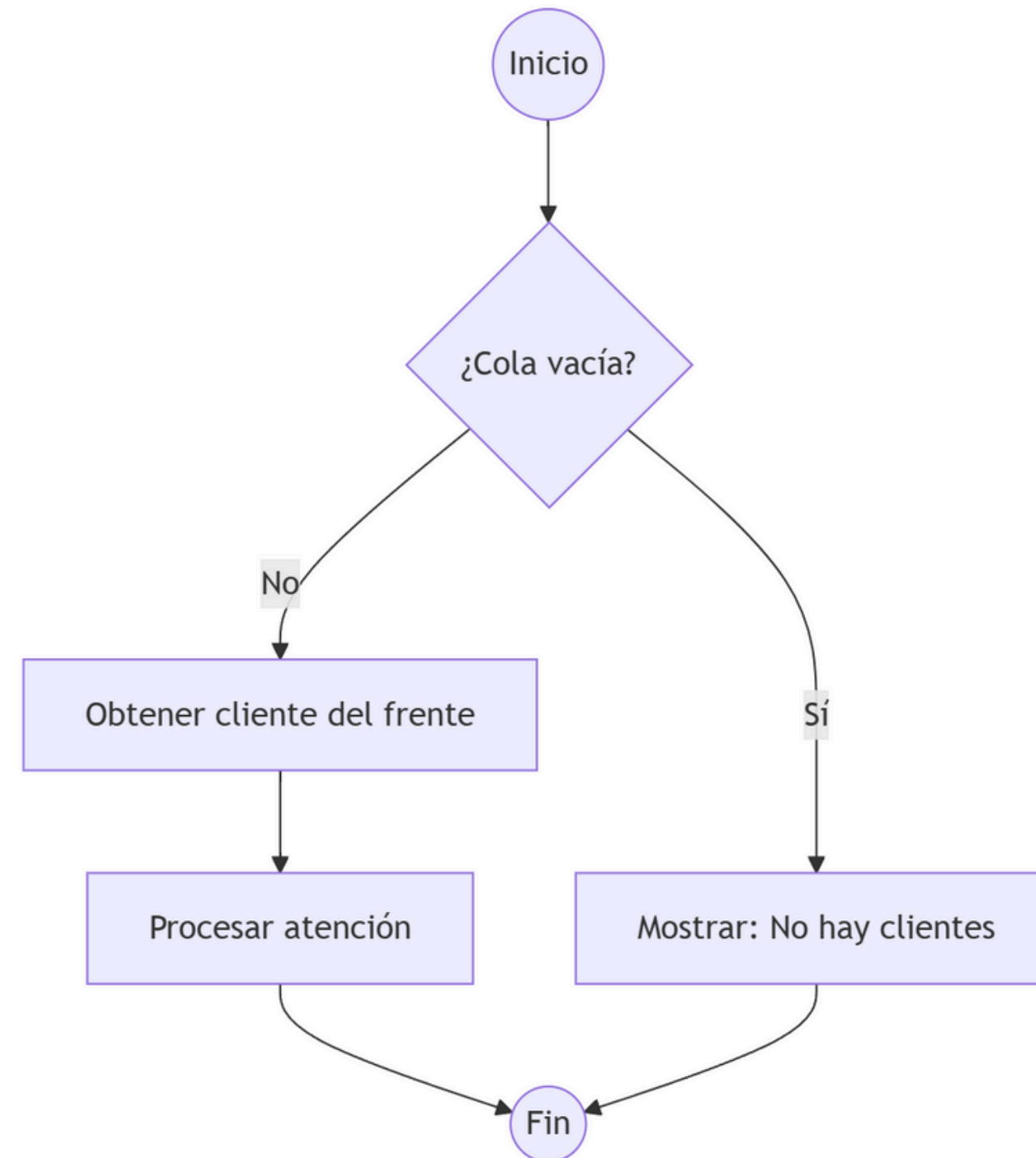
## Diagrama de Registro de clientes: lista enlazada





# Diagramas de Flujo

## Diagrama de Atender clientes (uso de colas)



# Plan de trabajo y roles asignados

## Fases del Proyecto:

- Fase 1: Apertura del Proyecto: Configuración de repositorios, planificación inicial y definición de roles.
- Fase 2: Diseño de Estructuras de Datos: Definición de listas, pilas, colas y validación de datos.
- Fase 3: Desarrollo de Funciones Críticas: Implementación de funciones de registro y gestión de clientes y transacciones.
- Fase 4: Integración y Pruebas Finales: Integración de módulos, pruebas de flujo completo y documentación.



# Plan de Trabajo para el Desarrollo del Proyecto FINPROC

- **Objetivo del Proyecto:**

Desarrollar un simulador que permita replicar y gestionar procesos bancarios, como el registro de clientes, transacciones y atención de clientes, utilizando estructuras de datos eficientes y técnicas ágiles.

- **Metodología de Trabajo:**

Utilización de Scrum adaptado para la gestión de tareas en tres sprints semanales, con seguimiento mediante herramientas colaborativas como GitHub y Obsidian (con complementos Kanban).



# **Roles y Responsabilidades del Equipo en FINPROC**

## **1. Gustavo Jhon Ccarita Velásquez – Líder de Proyecto:**

- **Responsabilidades:**
  - **Dirección general del proyecto.**
  - **Planificación, asignación de tareas y control de progreso semanal.**
  - **Desarrollo de funciones críticas.**
  - **Documentación del proyecto.**
  - **Supervisión del cumplimiento de los objetivos de cada sprint.**

## **2. Rodrigo – Desarrollador de Funcionalidades:**

- **Responsabilidades:**
  - **Desarrollo de funcionalidades como registro y visualización de datos (clientes y transacciones).**
  - **Realización de pruebas de validación.**
  - **Participación activa en el diseño y desarrollo de código.**

## **3. Sebastián – Desarrollador de Estructuras de Datos y Tester:**

- **Responsabilidades:**
  - **Implementación de estructuras de datos (listas, pilas, colas).**
  - **Realización de pruebas de validación (validación de transacciones y atención de clientes).**
  - **Mantenimiento de la documentación técnica.**

# Herramientas y Gestión del Proyecto

- **GitHub:** Repositorio central para el código, historial de commits, control de versiones.
- **Obsidian:** Gestión de tareas mediante Kanban (visualización del progreso).
- **Slack / Reuniones:** Comunicación constante y reuniones periódicas para revisión del avance.

**ENSIGNA**

**Muchas  
gracias**

