

Fatima Persi, Magali Pesqueira, Guillermina Cotrone

SISTEMA DE AUTOGESTION

ESTETICA DE UÑAS

OBJETIVO:

Crear un sistema para gestionar
todos los turnos, reservas, y
servicios de una estetica de uñas

ALCANCE DEL PRODUCTO

- **Acceso desde cualquier dispositivo con navegador.**
- **Consultar disponibilidad de turnos.**
- **Seleccionar:**
 - **Tipo de servicio (Kapping, Semi, Soft Gel)**
 - **Profesional (Gisela, Marisol, Valentina)**
- **Confirmar, cancelar o modificar reservas.**

MATRICES

```
def cargar_turnos():
    """Devuelve una lista con los turnos iniciales disponibles."""
    return [
        {"fecha": "2025-04-26", "hora": "10:00", "profesional": "Gisela", "servicio": "Kapping"},
        {"fecha": "2025-04-25", "hora": "14:00", "profesional": "Marisol", "servicio": "Semi"},
        {"fecha": "2025-04-26", "hora": "16:00", "profesional": "Valentina", "servicio": "Soft Gel"},
    ]
```

```
def mostrar_turnos(turnos):
    """Muestra en pantalla los turnos disponibles."""
    print("\nTurnos disponibles:")
    for i, turno in enumerate(turnos):
        print(f"{i + 1}. {turno['fecha']}{turno['hora']} - {turno['servicio']} con {turno['profesional']}")
```

```
def cancelar_turno(turnos, reservas):
    """Cancela una reserva a partir del nombre del cliente.

    Args:
        turnos (list): Lista de turnos disponibles.
        reservas (list): Lista de reservas existentes.

    Returns:
        tuple: Lista actualizada de turnos y reservas.
    """

    nombre = input("ingrese su nombre para cancelar turnos: ")
    nuevas_reservas = []
    turno_recuperado = None

    for r in reservas:
        if r["nombre"].lower() == nombre.lower():
            turno_recuperado = r["turno"]
            print("Turno cancelado correctamente")
        else:
            nuevas_reservas.append(r)

    if turno_recuperado:
        turnos.append(turno_recuperado)
    else:
        print("No se encontro una reserva con ese nombre.")
    return turnos, nuevas_reservas
```

```
def reservar_turnos(turnos, reservas):
    mostrar_turnos(turnos)
    try:
        opcion = int(input("\nElija el número del turno que desea reservar: ")) - 1
        turno = turnos[opcion]
    except (ValueError, IndexError):
        print("opcion invalida")
        return turnos, reservas

    nombre = input("ingrese su nombre: ")
    telefono = input("ingrese su telefono: ")

    cliente = {
        "nombre": nombre,
        "telefono": telefono,
        "turno": turno,
    }

    reservas.append(cliente)
    turnos = [t for t in turnos if t != turno]
    print(f"\nTurno reservado con éxito para {nombre}!")
    return turnos, reservas
```

LAMBDA

```
def filtrar_turnos(turnos, servicio=None, profesional=None):
    """Filtra la lista de turnos por servicio y/o profesional."""
    filtrados = list(filter(lambda t:
        (servicio is None or t['servicio'] == servicio) and
        (profesional is None or t['profesional'] == profesional),
        turnos
    ))
    return filtrados

def ver_nombre_clientes(reservas):
    """Muestra los nombres de los clientes que tienen una reserva.

    Args:
        reservas (list): Lista de reservas.
    """

    nombres = list(map(lambda r: r["nombre"], reservas))
    print("clientes con turnos reservados:")
    for nombre in nombres:
        print(f"- {nombre}")
```

MAP

```
def ver_nombre_clientes(reservas):  
    """Muestra los nombres de los clientes que tienen una reserva.  
  
    Args:  
        reservas (list): Lista de reservas.  
    """  
  
    nombres = list(map(lambda r: r["nombre"], reservas))  
    print("clientes con turnos reservados:")  
    for nombre in nombres:  
        print(f"- {nombre}")
```

REDUCE

```
def ver_resumen_reservas(reservas):  
    """Muestra un resumen de todas las reservas."""  
    print("\n--- Resumen de Reservas ---")  
    for r in reservas:  
        t = r["turno"]  
        print(f"{r['nombre']}: {t['fecha']} {t['hora']} - {t['servicio']} con {t['profesional']}")  
    print(f"\nTotal de reservas: {len(reservas)}")
```

FILTER

```
def filtrar_turnos(turnos, servicio=None, profesional=None):  
    """Filtra la lista de turnos por servicio y/o profesional."""  
    filtrados = list(filter(lambda t:  
        (servicio is None or t['servicio'] == servicio) and  
        (profesional is None or t['profesional'] == profesional),  
        turnos  
    ))  
    return filtrados
```

LISTAS

```
turnos = [t for t in turnos if t != turno]
```


STRING

Usamos f-strings para que los mensajes salgan bien formateados. Todas utilizaron f-strings para mostrar información al usuario de forma clara

```
print(f"{i + 1}. {turno['fecha']}{turno['hora']} - {turno['servicio']} con {turno['profesional']}")
```

DOCSTRING

Es un comentario entre triples comillas que se coloca al inicio de una función para explicar que hace

```
"""Devuelve una lista con los turnos iniciales disponibles."""
```

=== MENÚ PRINCIPAL ===

1. Ver turnos disponibles
2. Filtrar turnos
3. Reservar turno
4. Cancelar turno
5. Ver resumen de reservas
6. Ver nombres de clientes
7. Salir

Seleccione una opcion:

=== MENÚ PRINCIPAL ===

1. Ver turnos disponibles
2. Filtrar turnos
3. Reservar turno
4. Cancelar turno
5. Ver resumen de reservas
6. Ver nombres de clientes
7. Salir

Seleccione una opcion: 2

Filtrar por servicio (Kapping, Semi, Soft Gel o ENTER para todos): Semi

Filtrar por profesional (Gisela, Marisol, Valentina o ENTER para todas): Valentina

Turnos disponibles:

Seleccione una opcion: 1

Turnos disponibles:

1. 2025-04-25 14:00 - Semi con Marisol
2. 2025-04-26 16:00 - Soft Gel con Valentina

Turnos disponibles:

1. 2025-04-25 14:00 - Semi con Marisol
2. 2025-04-26 16:00 - Soft Gel con Valentina

Elija el número del turno que desea reservar: 1
ingrese su nombre: Fatima
ingrese su telefono: 2227584761

Turno reservado con éxito para Fatima!

--- Resumen de Reservas ---

juana: 2025-04-26 10:00 - Kapping con Gisela

Total de reservas: 1

Seleccione una opcion: 4

ingrese su nombre para cancelar turnos: Fatima
Turno cancelado correctamente



MUCHAS
GRACiAS

