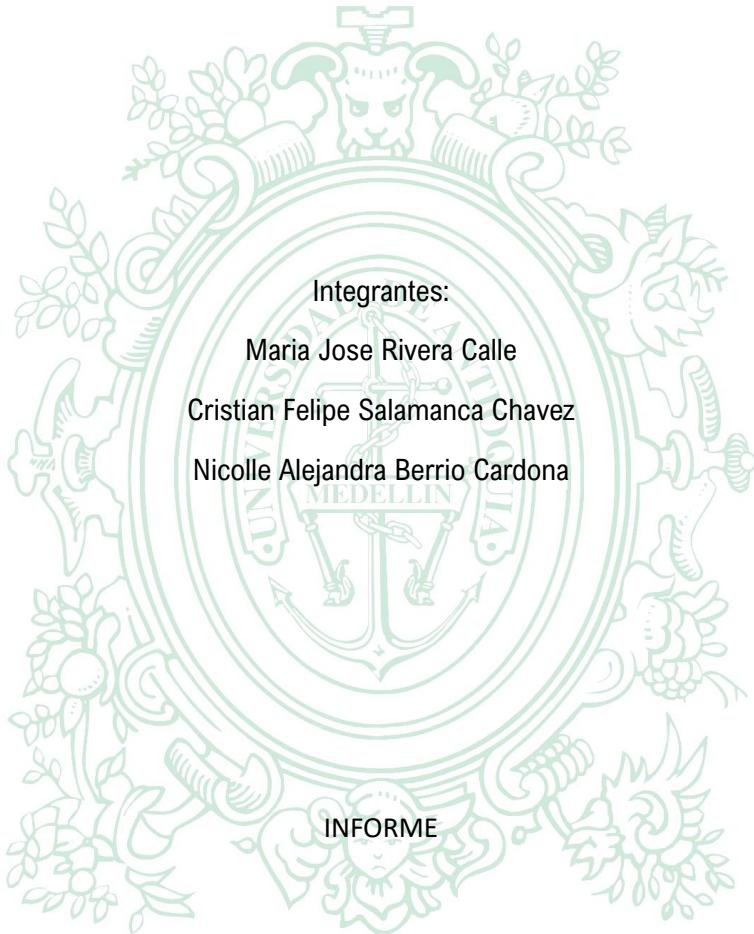


TRABAJO FINAL ALGORITMIA Y PROGRAMACIÓN



Integrantes:

Maria Jose Rivera Calle

Cristian Felipe Salamanca Chavez

Nicolle Alejandra Berrio Cardona

INFORME

UNIVERSIDAD DE ANTIOQUIA

1 803
ENTREGA 1

UNIVERSIDAD DE ANTIOQUIA

FACULTAD DE INGENIERIA

1. Reporte de visión

El software **Resort Waira** es una aplicación de consola diseñada para gestionar de manera eficiente la atención a los clientes dentro del resort. Su propósito es simular y organizar las actividades relacionadas con la planeación de la demanda, el registro de clientes, el control de recursos disponibles, y la generación de reportes administrativos.

Los **objetivos principales** del sistema son:

- Facilitar el proceso de registro de clientes garantizando datos correctos y completos.
- Automatizar los cálculos de costos, ventas y ganancias para optimizar la gestión del resort.
- Proveer al administrador herramientas que permitan consultar en tiempo real la disponibilidad de servicios, habitaciones, alimentación y turismo.
- Permitir la exportación de los datos en un archivo CSV para conservar los registros y facilitar futuros análisis.

Los **beneficios esperados** incluyen:

- Mayor organización en la gestión de clientes y recursos.
- Reducción de errores humanos mediante validaciones automáticas.
- Acceso rápido y seguro a reportes clave para la toma de decisiones.
- Un entorno sencillo y amigable que puede ser usado por cualquier miembro del resort.

2. Especificación de requisitos

Requisitos funcionales

1. **Planeación de la demanda:** el sistema debe permitir configurar la disponibilidad inicial de habitaciones, alimentación, turismo y otros servicios.
2. **Registro de clientes:** debe registrar clientes individuales, parejas o familias, validando nombre, apellido y documento según reglas de longitud y tipo de datos.
3. **Control de mascotas:** debe llevar un conteo de las mascotas registradas en el resort.
4. **Cálculo de costos, ventas y ganancias:** el sistema debe calcular automáticamente los valores según la demanda registrada y los costos fijos establecidos.
5. **Módulo de administración:** acceso restringido por usuario y contraseña, donde se podrá consultar:
 - Total de clientes en el hotel.
 - Total de clientes por tipo (individual, pareja, familia).
 - Total de mascotas.

- Disponibilidad de habitaciones, alimentación y turismo.
 - Reportes de ventas, costos y ganancias.
6. **Exportar datos:** al finalizar el día, el usuario podrá exportar la lista de clientes y los resultados generales en formato CSV.
7. **Salir del sistema:** el programa debe permitir cerrar la jornada de manera ordenada.

Requisitos no funcionales

- **Usabilidad:** interfaz de consola clara y sencilla, con menús intuitivos y mensajes explicativos.
- **Seguridad:** acceso al módulo de administración protegido por credenciales.
- **Compatibilidad:** ejecutable en cualquier sistema operativo que soporte Python 3.x.
- **Rendimiento:** el sistema debe procesar rápidamente registros y cálculos sin retrasos perceptibles.
- **Fiabilidad:** los datos deben almacenarse correctamente en estructuras internas y exportarse sin pérdida de información.
- **Mantenibilidad:** el código debe estar organizado y documentado para facilitar futuras mejoras.

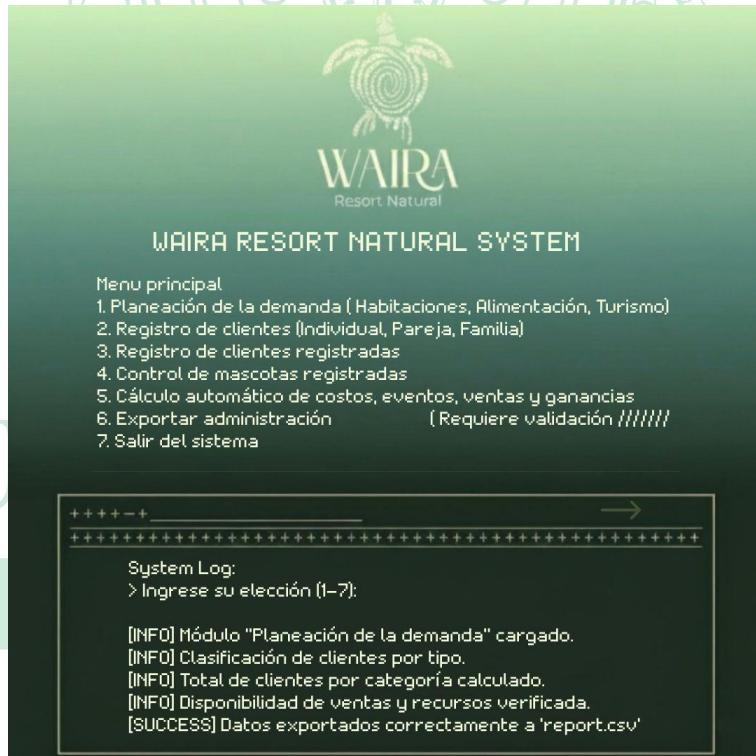
Librerías

Para el desarrollo del software **Resort Waira**, se consideran necesarias las siguientes librerías de Python:

- **csv (estándar de Python)**
 - Uso: Exportar e importar datos en archivos CSV.
 - Aplicación: Guardar la información de clientes y reportes de ventas, costos y ganancias al final de la jornada.
- **datetime (estándar de Python)**
 - Uso: Manejo de fechas y horas.
 - Aplicación: Registrar la fecha y hora de ingreso de clientes, y asociar los reportes diarios a un día específico.
- **getpass (estándar de Python)**
 - Uso: Entrada de contraseñas de manera segura.
 - Aplicación: Proteger el acceso al módulo de administración sin mostrar la contraseña en pantalla.

- **re (expresiones regulares, estándar de Python)**
 - Uso: Validar cadenas de texto según patrones.
 - Aplicación: Comprobar que nombres y apellidos no tengan números, y que los documentos sean únicamente numéricos y con la longitud permitida.
- **tabulate**
 - Uso: Mostrar datos en tablas organizadas en la consola.
 - Aplicación: Presentar reportes administrativos (clientes registrados, disponibilidad, ventas) de manera más clara y legible.
- **rich**
 - Uso: Mejorar la visualización en consola con colores y tablas.
 - Aplicación: Dar un formato más amigable a los menús y reportes, facilitando la interacción del usuario.

Visual



Algoritmos

Pseudocódigo General del Sistema

Inicio

Mostrar "Bienvenido a Resort Waira"

Mostrar menú principal:

1. Planeación de la demanda
2. Registro de clientes
3. Cálculo de costos, ventas y ganancias
4. Módulo de administración
5. Exportar datos
6. Salir

Leer opción

Segun opcion hacer

- CASO 1: Ejecutar Planeación de la Demanda
- CASO 2: Ejecutar Registro de Clientes
- CASO 3: Ejecutar Cálculo de Costos
- CASO 4: Ejecutar Módulo de Administración
- CASO 5: Ejecutar Exportación de Datos
- CASO 6: Salir del sistema

Fin segun

Fin|

Planeación de la Demanda

Algoritmo PlaneacionDemandada

Var entero: habitaciones, alimentacion, turismo

Inicio

Escribir "Configurar disponibilidad de habitaciones"

Leer habitaciones

Escribir "Configurar disponibilidad de alimentación"

Leer alimentacion

Escribir "Configurar disponibilidad de turismo"

Leer turismo

Escribir "Planeación registrada correctamente"

Fin|

Registro de Clientes



DA
UIA

```

Algoritmo RegistroCientes
Var string: nombre, apellido, tipoCliente
Var entero: documento, mascotas

Inicio
    Escribir "Ingrese nombre"
    Leer nombre
    Escribir "Ingrese apellido"
    Leer apellido
    Escribir "Ingrese documento"
    Leer documento

    Escribir "Seleccione tipo de cliente
(Individual/Pareja/Familia)"
    Leer tipoCliente

    Escribir "¿Desea registrar mascotas? (1=Sí, 0=No)"
    Leer mascotas

    Si mascotas > 0 Entonces
        Escribir "Número de mascotas registradas: ", mascotas
    Fin Si

    Escribir "Registro completado"
Fin|

```

Cálculo de Costos

```

Algoritmo CalculoCostos
Var real: costos, ventas, ganancias

```

```

Inicio
    Escribir "Ingrese costos totales"
    Leer costos

    Escribir "Ingrese ventas generadas"
    Leer ventas

    ganancias <- ventas - costos

    Escribir "Reporte:"
    Escribir "Costos: ", costos
    Escribir "Ventas: ", ventas
    Escribir "Ganancias: ", ganancias
Fin|

```

DAD
QUIA

Administración

```
Algoritmo Administracion
Var string: usuario, clave
Var entero: intentos

Inicio
    intentos <- 0

    Mientras intentos < 3 Hacer
        Escribir "Ingrese usuario"
        Leer usuario
        Escribir "Ingrese contraseña"
        Leer clave

        Si usuario = "admin" Y clave = "1234" Entonces
            Escribir "Acceso concedido"
            Escribir "Mostrar reportes: clientes, mascotas,
                disponibilidad, ventas"
            intentos <- 3
        Sino
            Escribir "Acceso denegado"
            intentos <- intentos + 1
        Fin Si
    Fin Mientras
Fin|
```

Exportar Datos

```
Algoritmo ExportarDatos
Inicio
    Escribir "Exportando lista de clientes..."
    Escribir "Exportando reportes financieros..."
    Escribir "Exportación exitosa en formato CSV"
Fin|
```

Salida

```
Algoritmo Salida
Inicio
    Escribir "Cerrando sistema..."
    Escribir "Gracias por usar Waira Resort"
Fin|
```

Estructuras de Datos

Estructuras de Datos	Uso principal en el sistema	Ejemplo de datos	Justificación
Listas	Manejo dinámico y ordenado de colecciones como clientes, registros de demanda, reservas y ventas.	clientes = [{"nombre":"Ana", "apellido":"Torres", "documento":"123456789"}, {"nombre":"Luis", "apellido":"Ramírez", "documento":"987654321"}] reportes = [{"fecha":"2025-10-01", "clientes":25, "ventas":1500000, "ganancias":600000}, {"fecha":"2025-10-02", "clientes":32, "ventas":1800000, "ganancias":700000}]	El uso de listas permite agregar o eliminar clientes conforme llegan o se van, y facilita la iteración para realizar validaciones, cálculos o generar reportes. Además, es posible ordenar o filtrar esta lista según los criterios que se requieran, como tipo de cliente o fecha de llegada.
Diccionarios	Los diccionarios se emplearán para almacenar información con clave-valor, lo que ofrece un acceso rápido y directo a los datos, sin necesidad de recorrer una estructura completa. Esto es especialmente útil para manejar parámetros configurables y atributos específicos como costos por tipo de habitación, disponibilidad de servicios, y configuraciones	costos = {"habitacion_individual":100000, "habitacion_pareja":180000, "habitacion_familia":250000, "costo_alimentacion":30000} disponibilidad = {"habitacion_individual":10, "habitacion_pareja":5, "habitacion_familia":3, "mascotas_permitidas":15} usuarios = {"admin":"1234", "gerente":"abcd"}	Esto permite que el sistema pueda actualizar rápidamente la disponibilidad o los precios sin afectar el resto de la estructura y con acceso eficiente durante la ejecución de cálculos y generación de reportes.

	generales del sistema.		
Tuplas	Las tuplas se usarán para almacenar datos constantes o inmutables, como tarifas fijas o parámetros generales que no deben cambiar durante la ejecución del programa. Esto ayuda a proteger la integridad de estos datos y a optimizar el uso de memoria.	<pre> tarifas = (100000,180000,250000) # Individual, Pareja, Familia tipos_cliente = ("Individual","Pareja","Familia") servicios_basicos = ("Habitación","Alimentación","Turismo") </pre>	Al ser inmutables, las tuplas garantizan que estos valores no sean modificados accidentalmente, contribuyendo a la estabilidad y seguridad del sistema.

UNIVERSIDAD DE ANTIOQUIA

1 8 0 3