**UNIVERSIDAD PRIVADA ANTENOR ORREGO**

**FACULTAD DE INGENIERÍA**

**PROGRAMA DE ESTUDIOS DE INGENIERÍA DE COMPUTACIÓN Y SISTEMAS**

****

**Proyecto:**

Sistemas de reserva de un Hotel

**Integrantes:**

Choncen Gutierrez Daniela

Cuba Prieto Mauricio

Morales Robles Jeancarlo

Rubiños Ángulo Fabrizio

**Docente:**

[SILVIA ANA RODRIGUEZ AGUIRRE](https://upao.instructure.com/courses/30498/users/408)

**Curso:**

Algoritmia y Programación

**Horario:**

10:40 a.m. – 2:15p.m

**Trujillo Perú**

**2023**

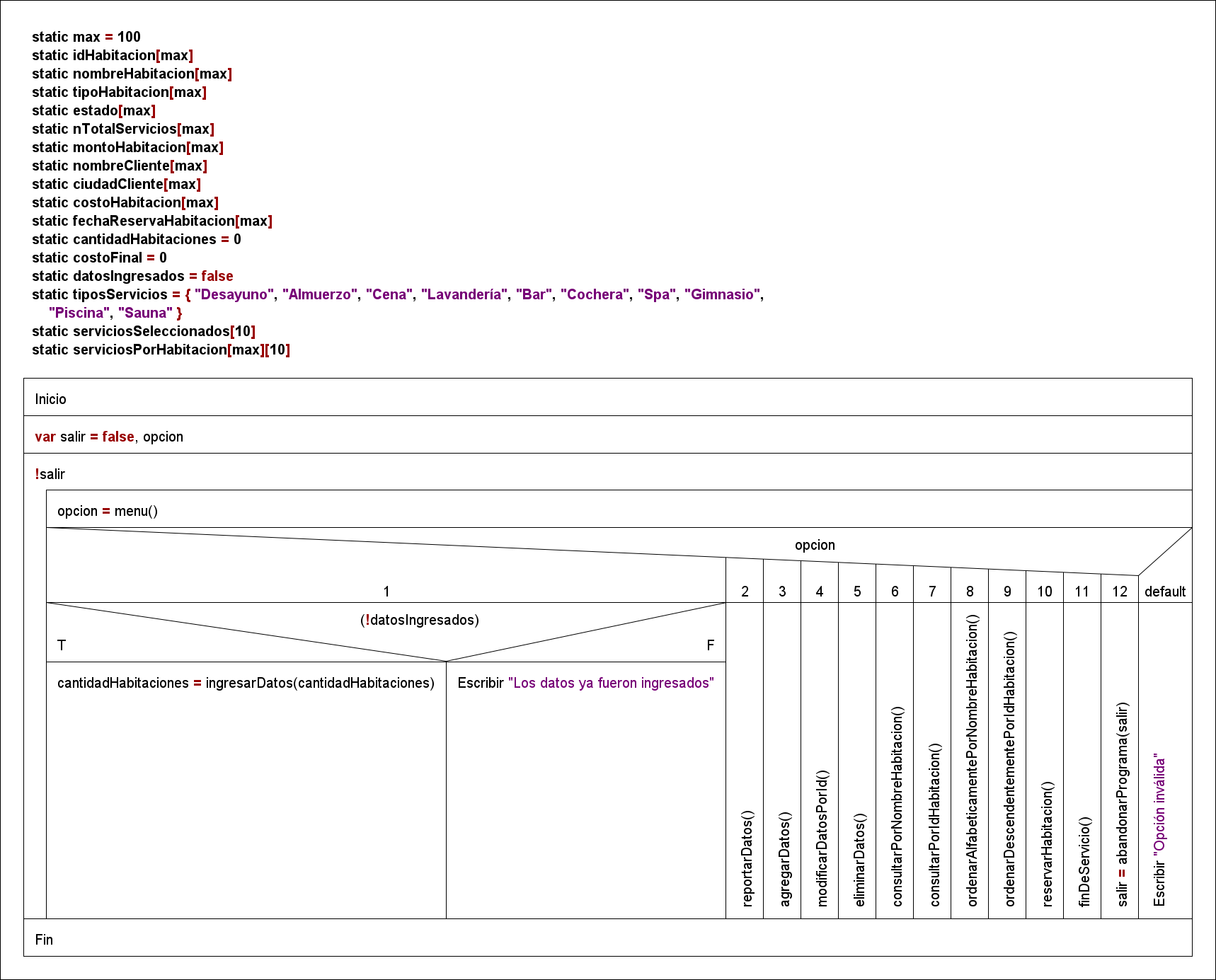
**Esquema de Flujo de Datos:**

idHabitacion[], nombreHabitacion[], tipoHabitacion[], estado[], nTotalServicios[], montoHabitacion[], nombreCliente[], ciudadCliente[], costoHabitacion[], fechaReservaHabitacion[], costoFinal, tiposServicios[], serviciosSeleccionados[], serviciosPorHabitacion[]

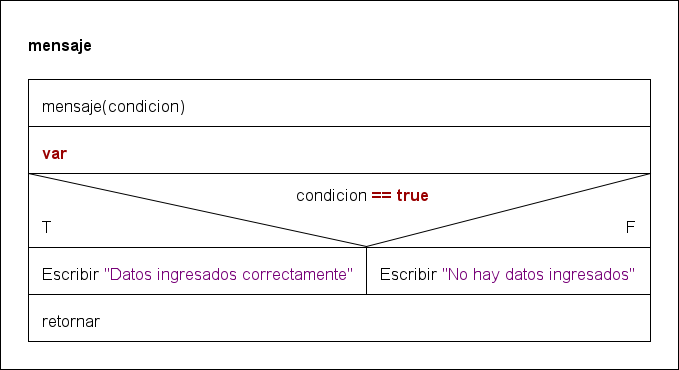
opción, idHabitacion[], nombreHabitacion[], tipoHabitacion[], nTotalServicios[], nombreCliente[], ciudadCliente[], fechaReservaHabitacion[], serviciosSeleccionados[], serviciosSeleccionados[], serviciosPorHabitacion[][], cantidadHabitaciones

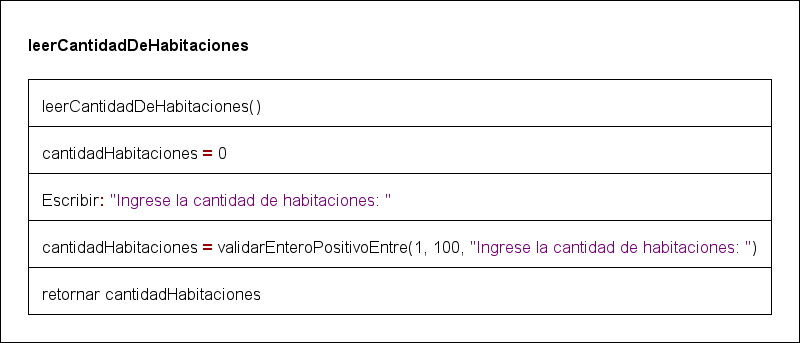
**Proceso**

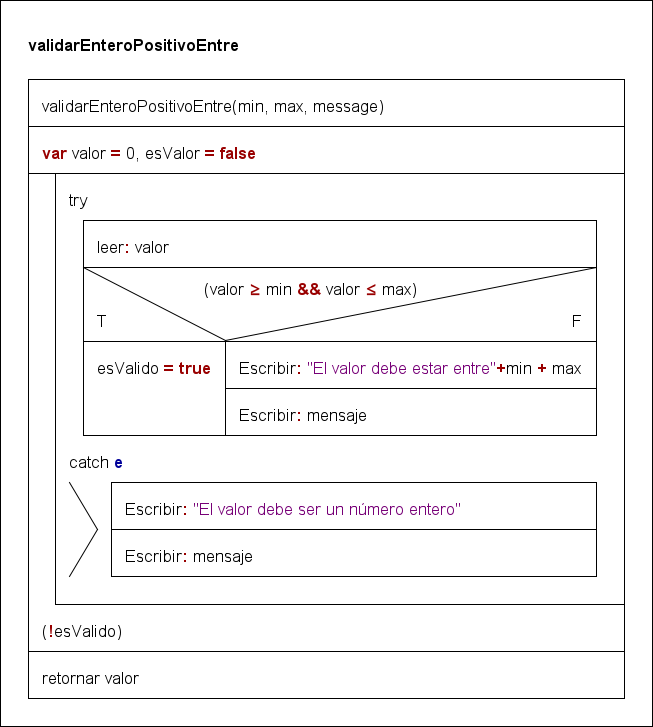
**Diagramas N-S**

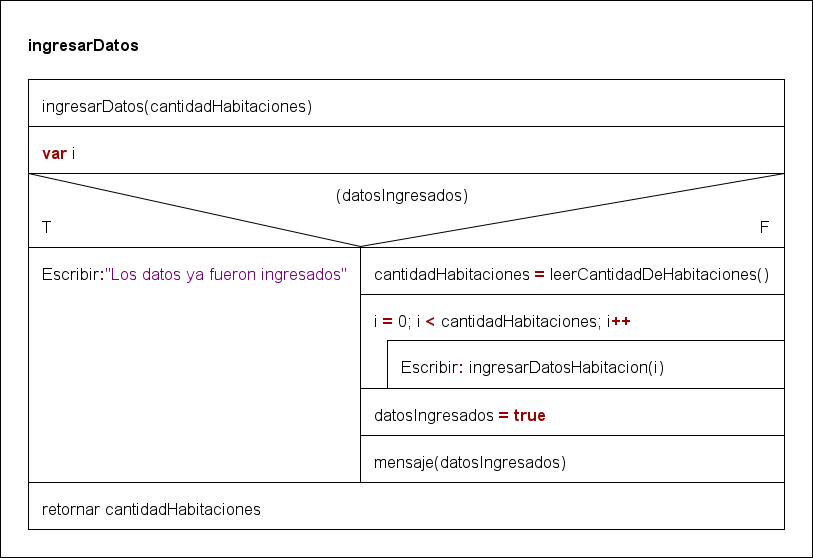


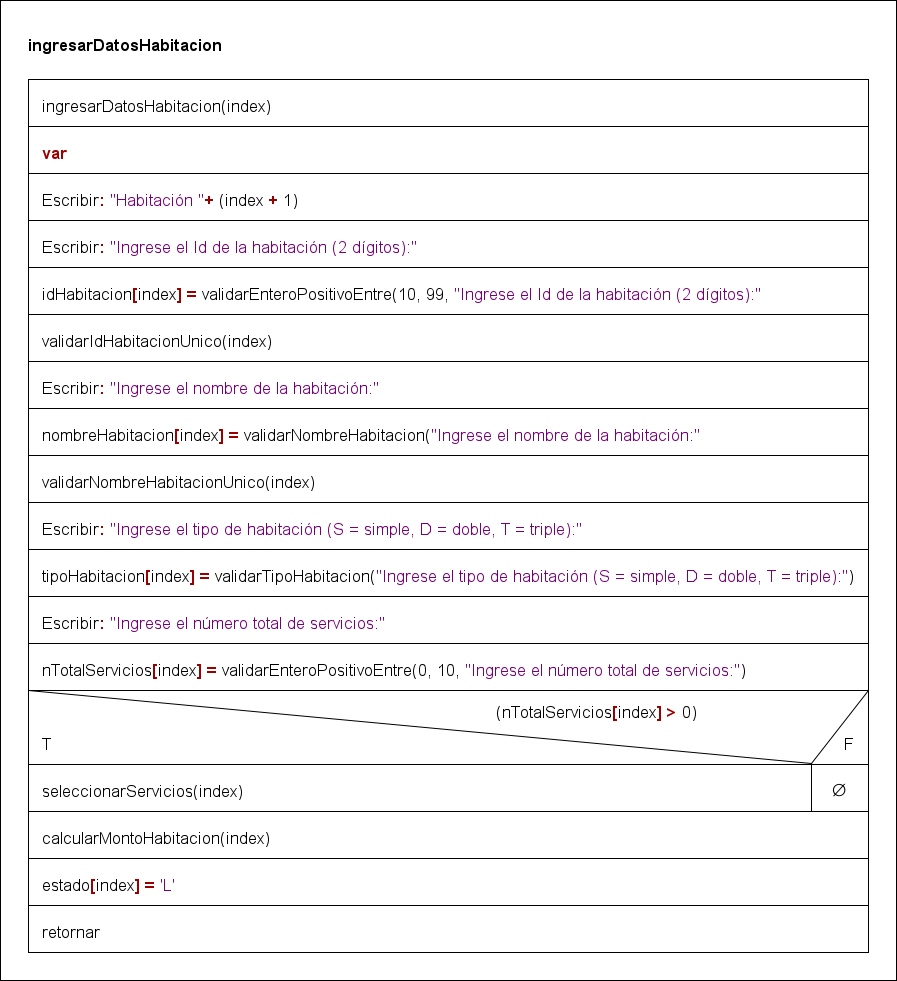


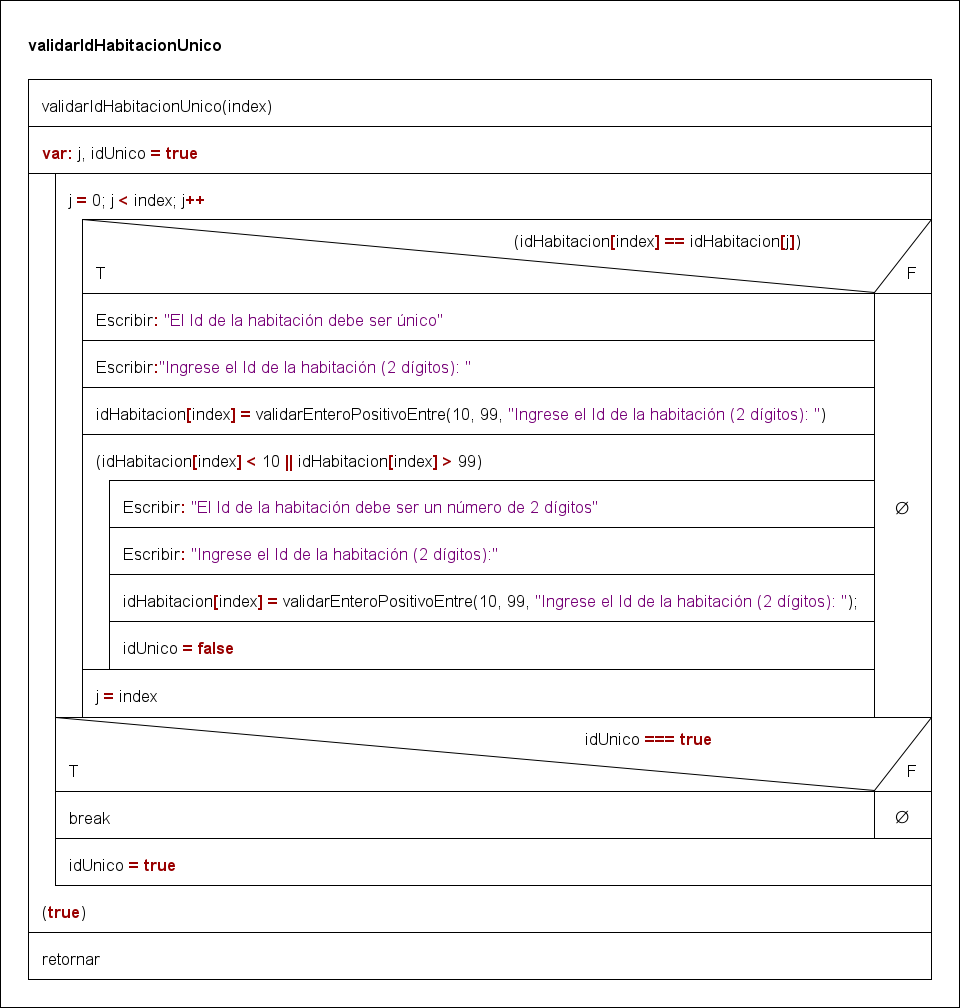


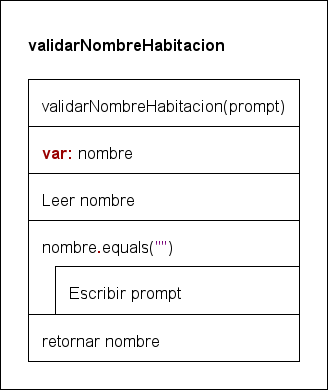
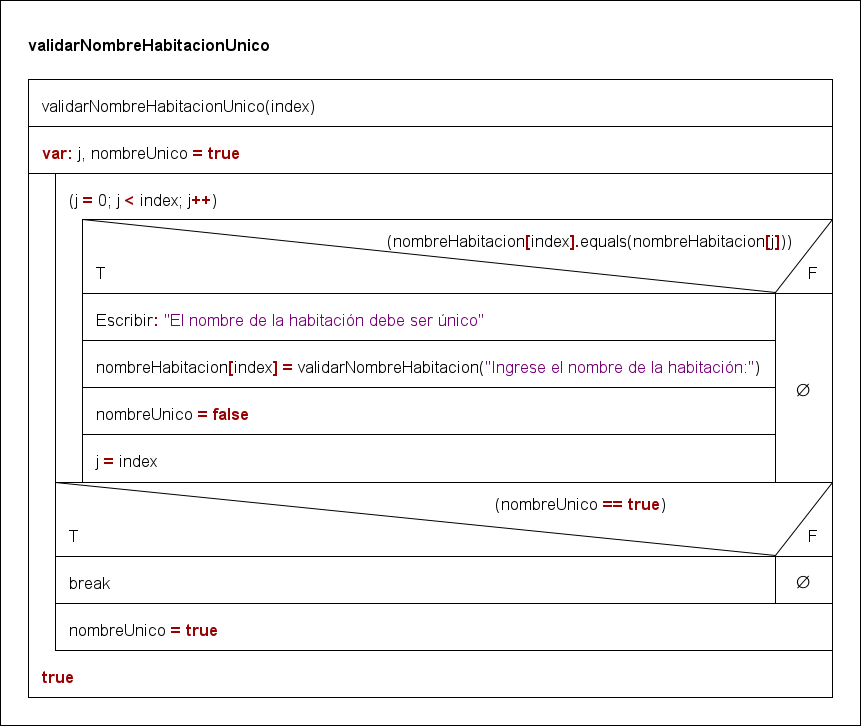


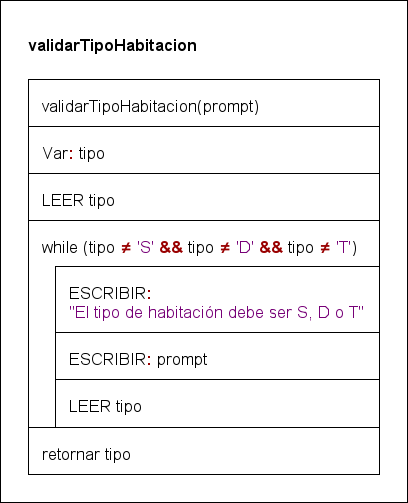




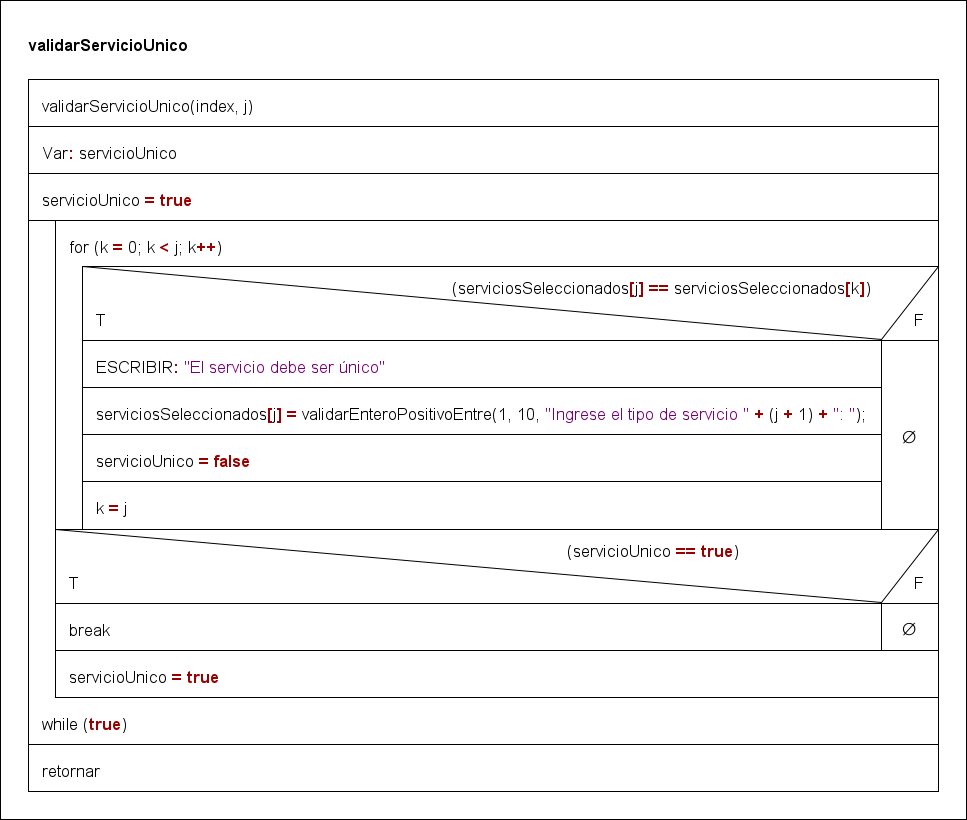


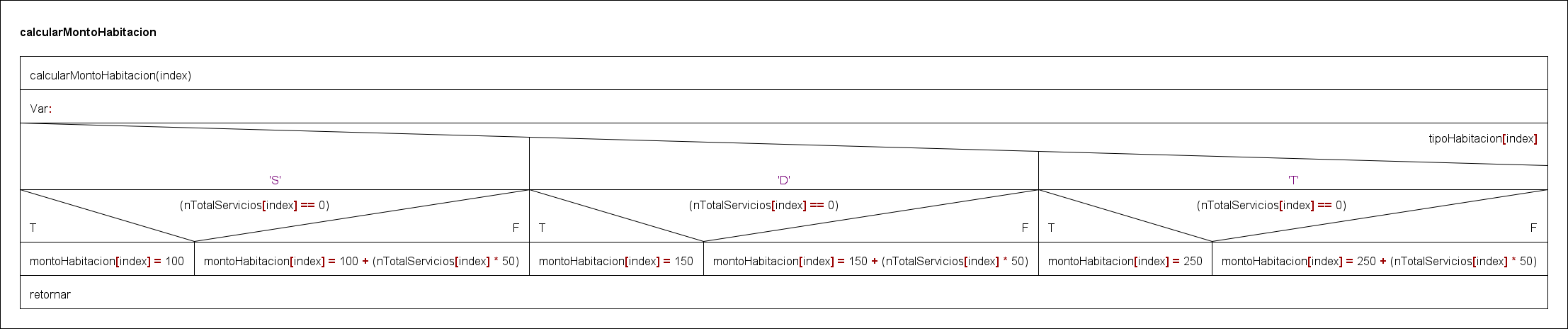


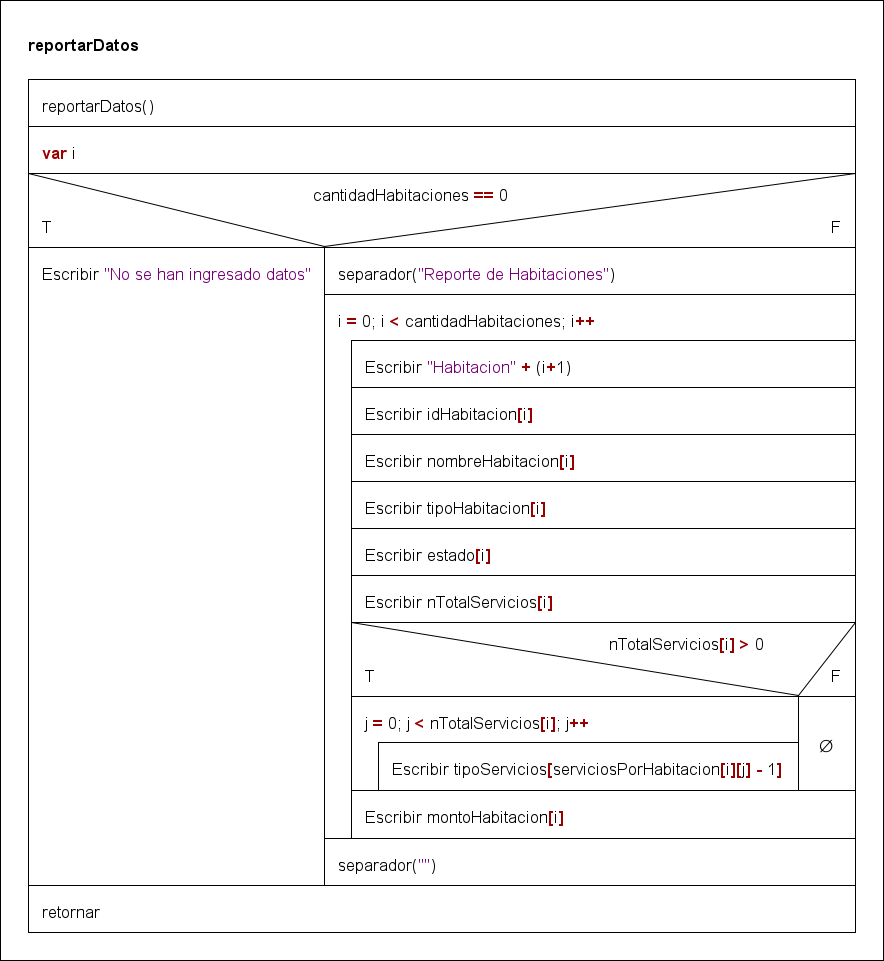


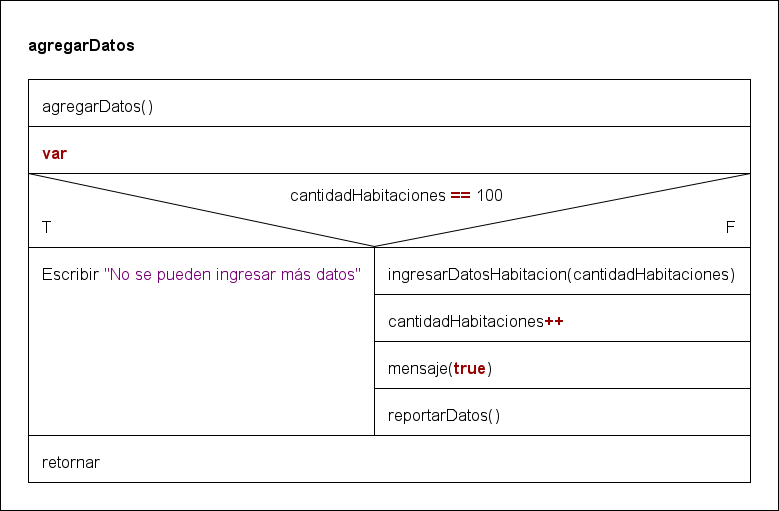


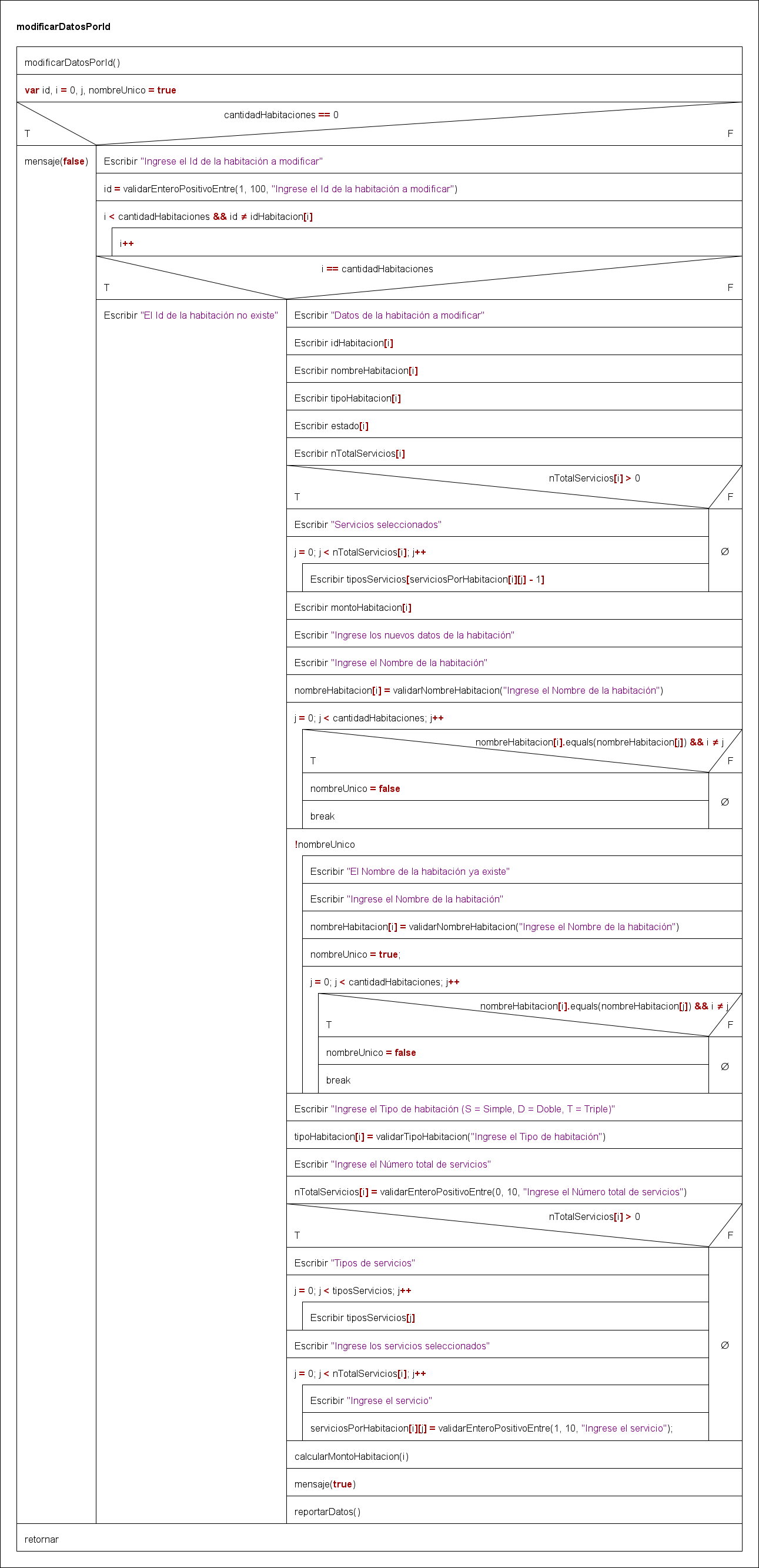


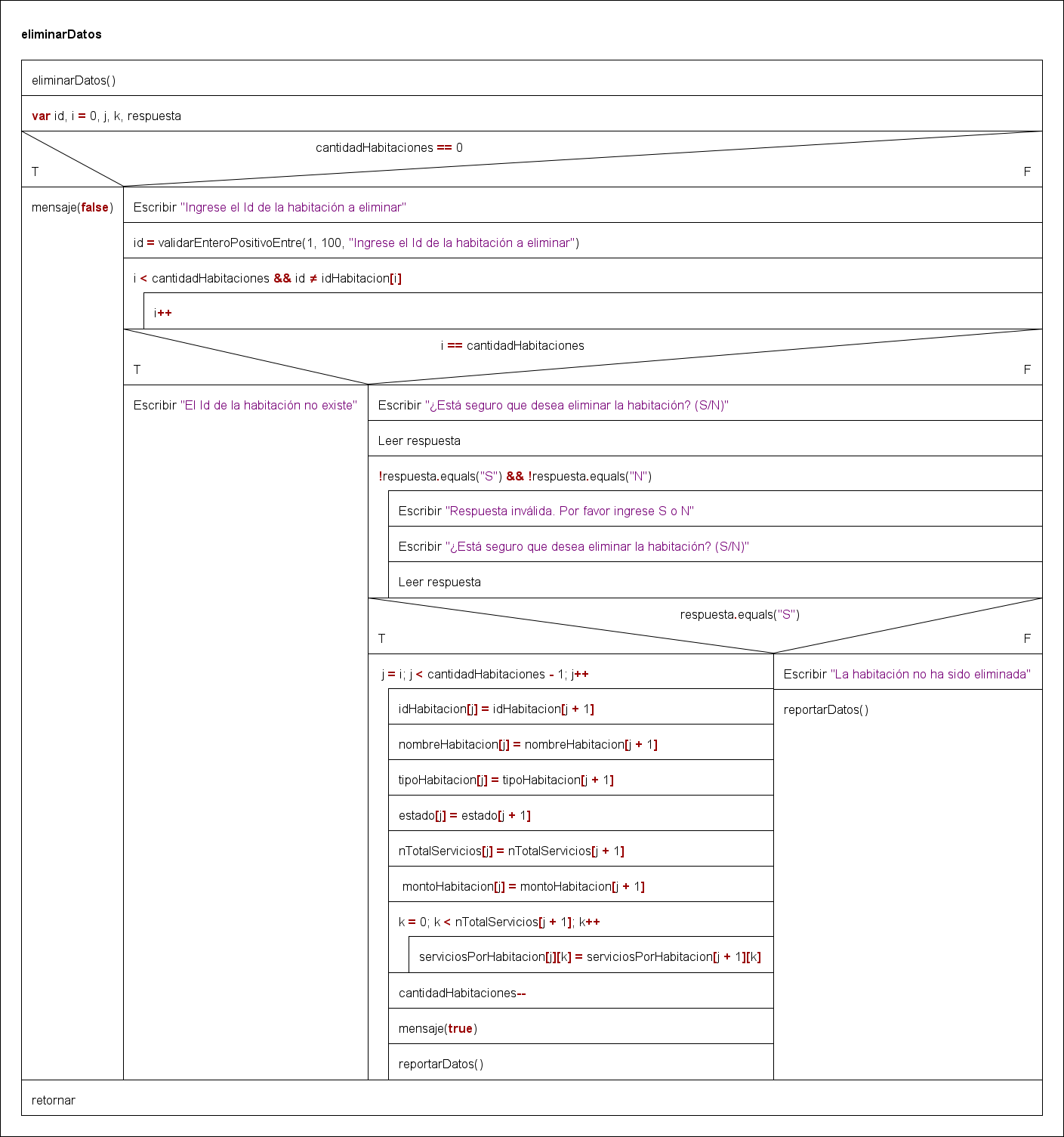


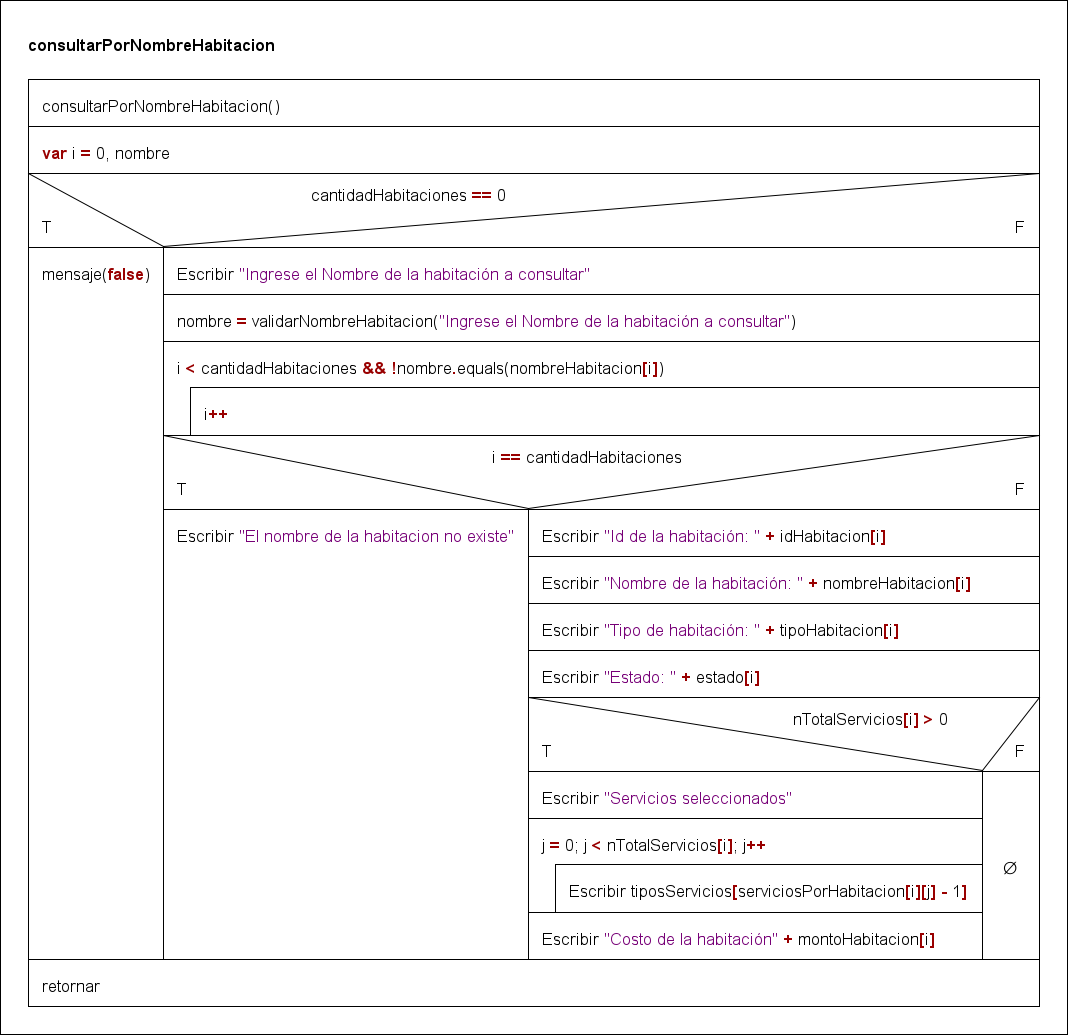


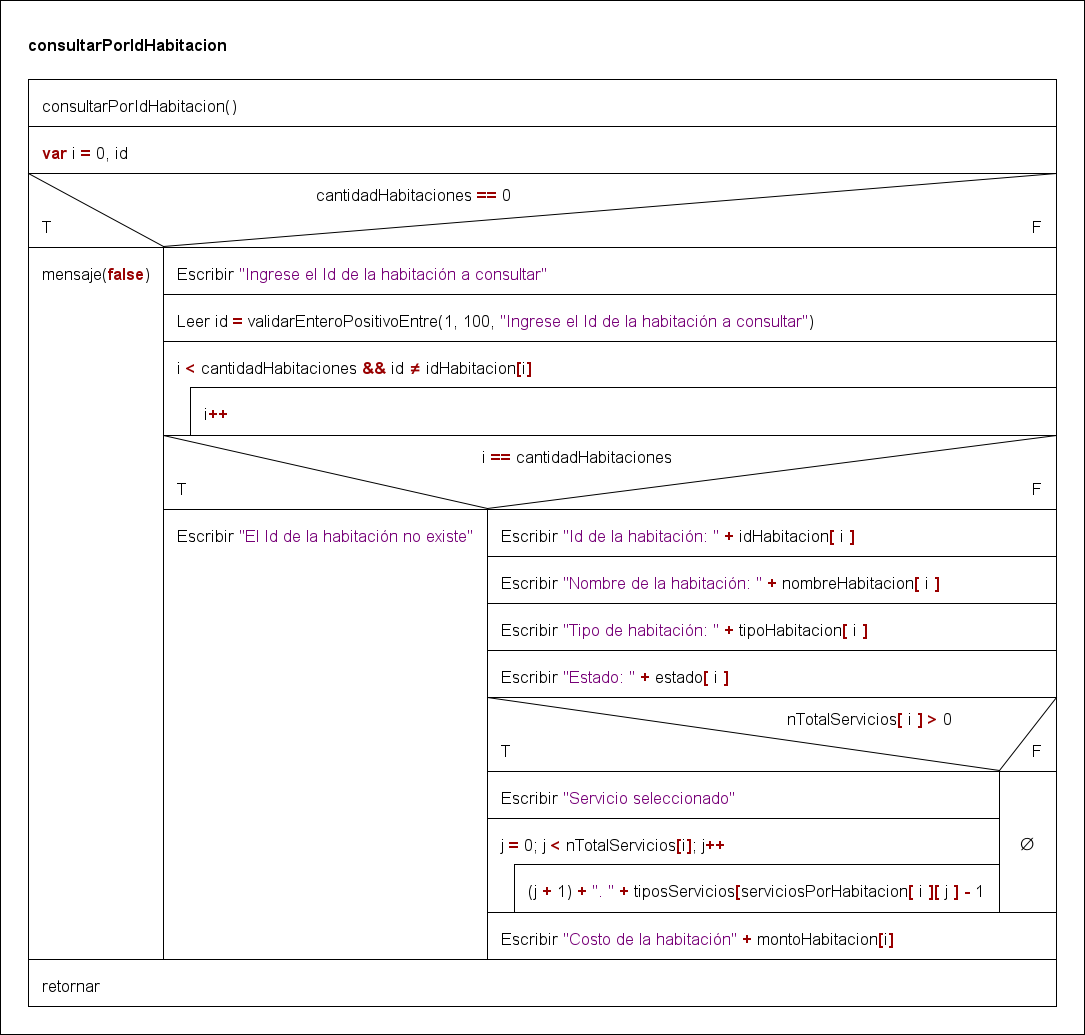


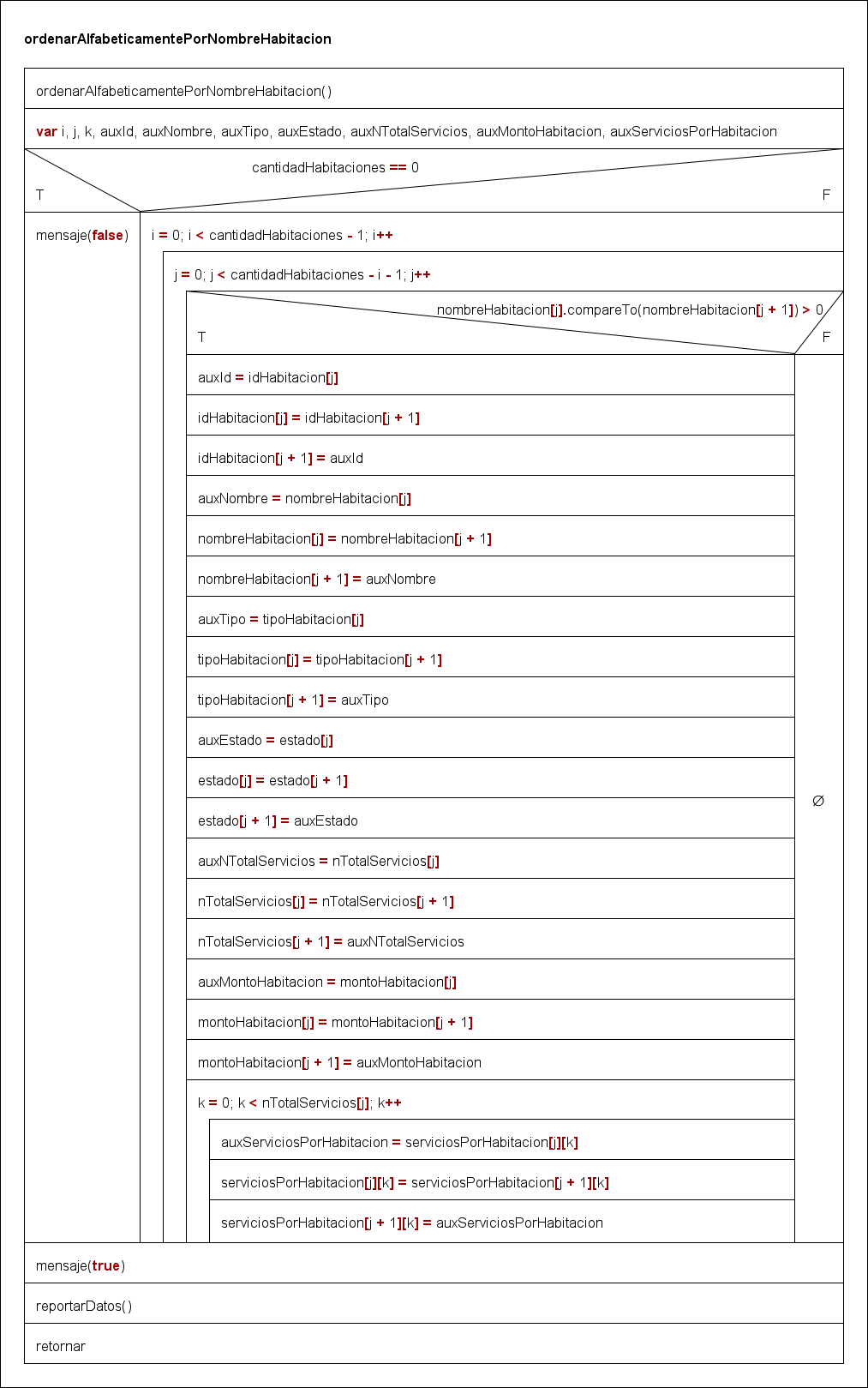


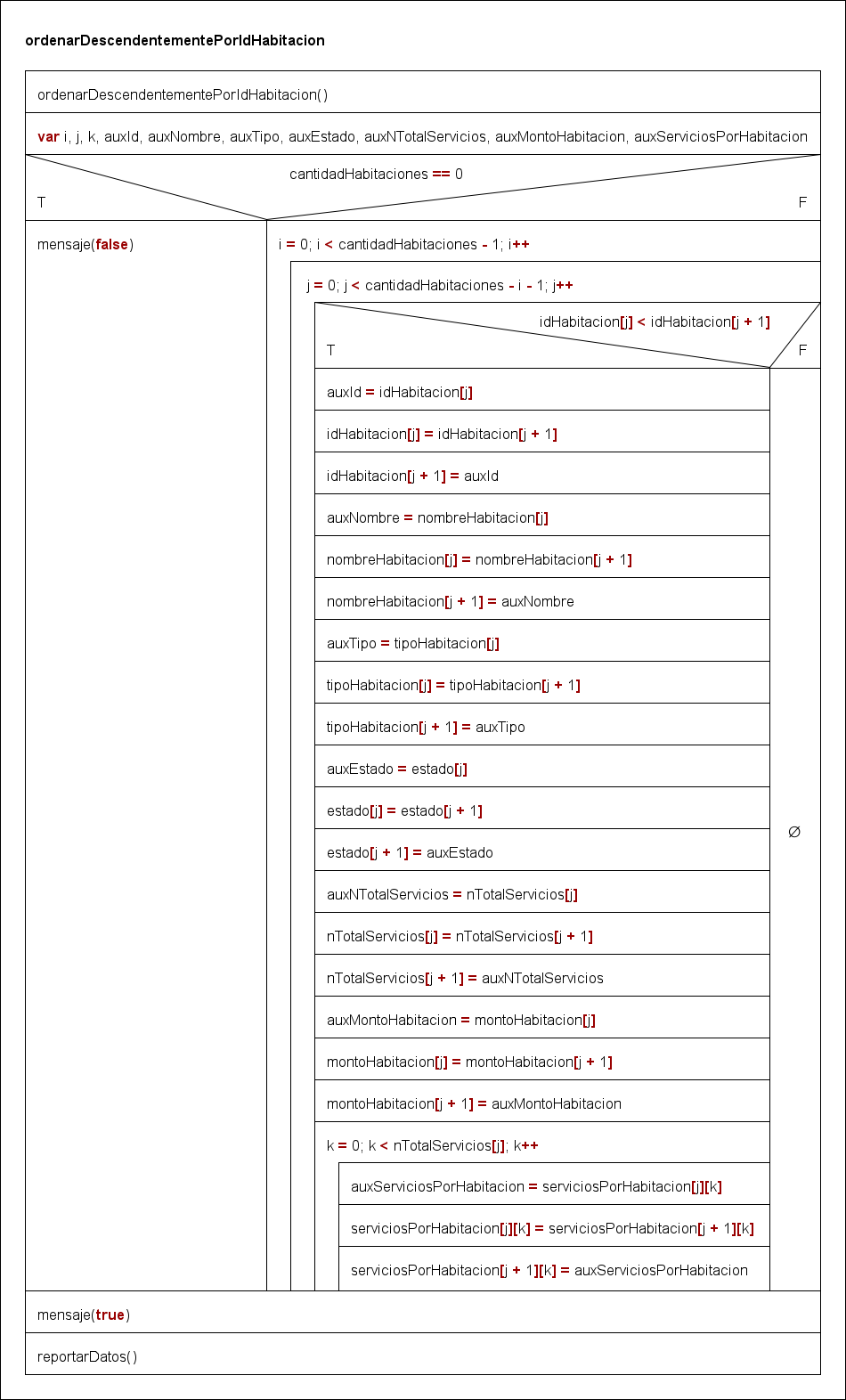


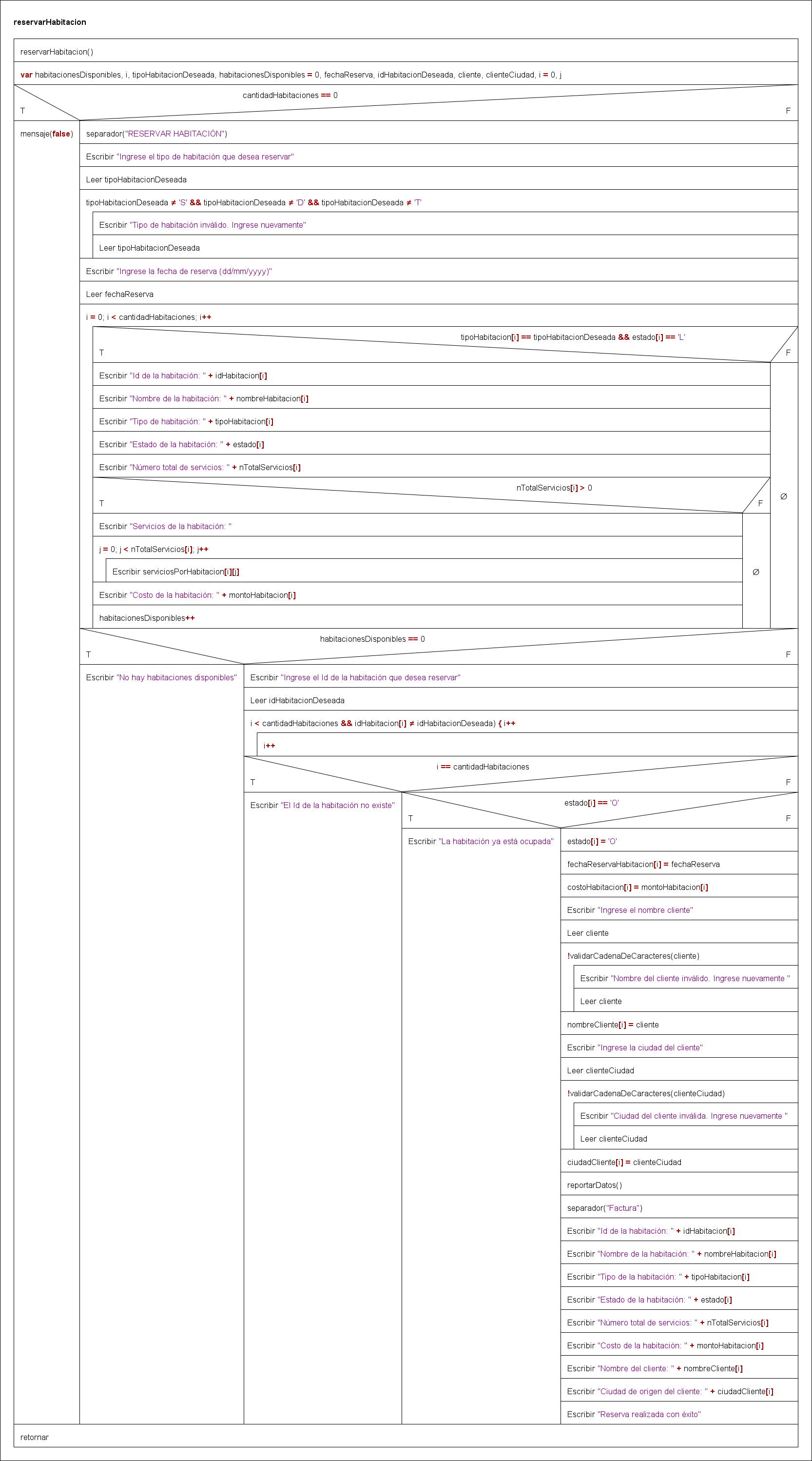


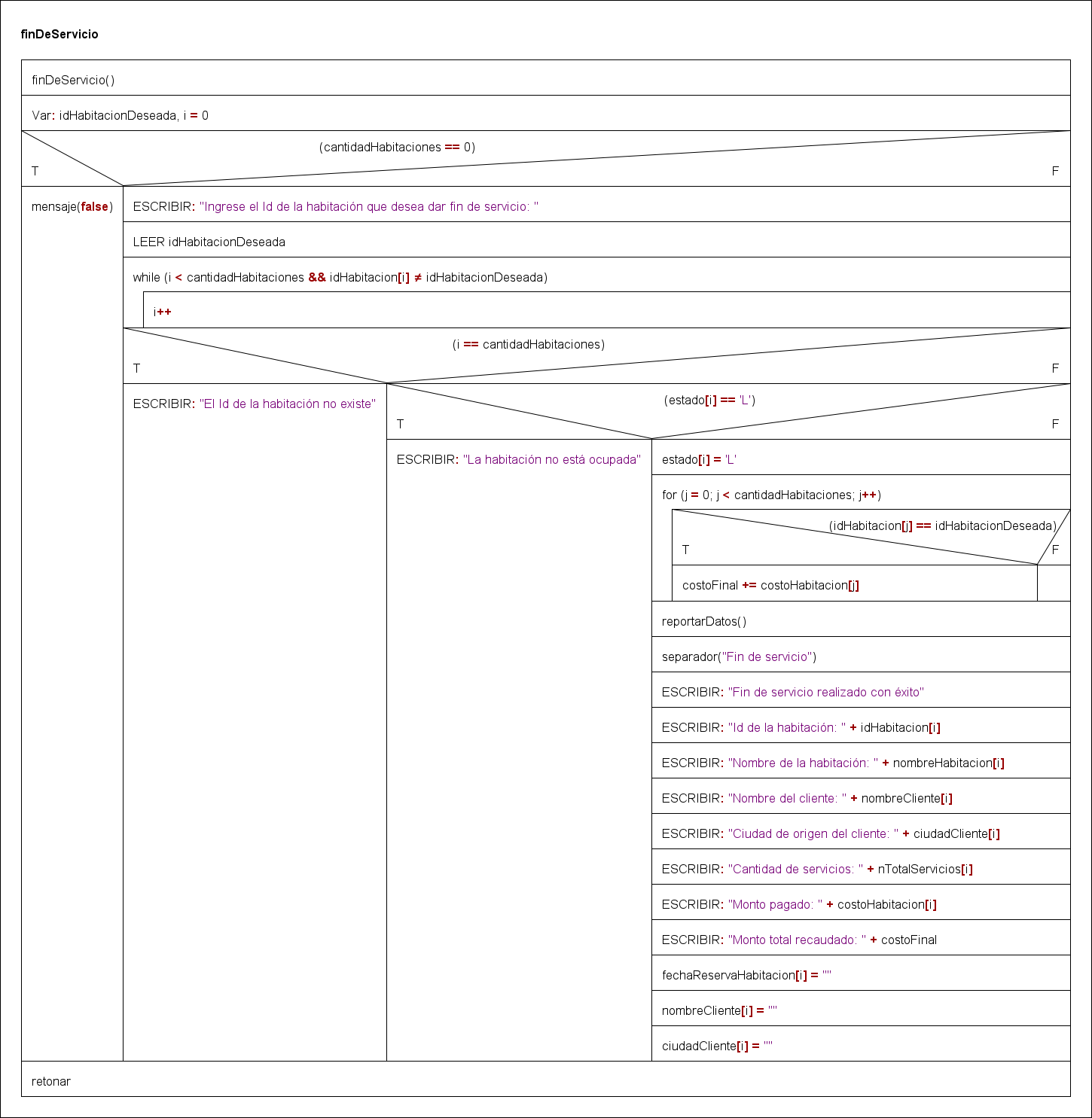
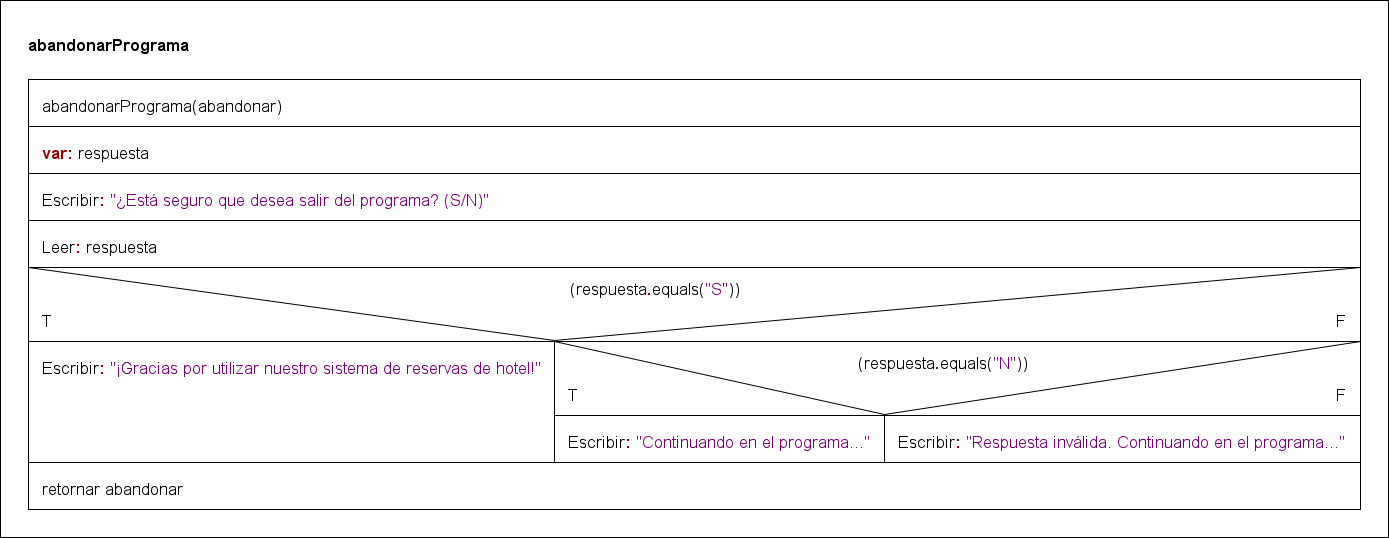


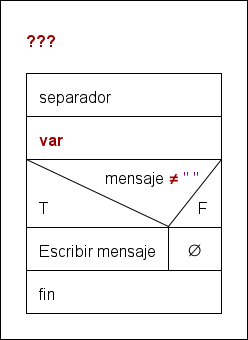












Código en java:

*/\* Sistema de reserva de un hotel \*/*

*/\**

*Implementar un sistema de reserva de habitaciones de un hotel. El sistema debe permitir:*

*- Ingresar datos de N habitaciones de hotel (se realiza una sola vez) IdHabitación (int 2 dígitos), Nombre Habitación (String), Tipo Habitación (char) ‘S’=simple, ‘D’=doble, ‘T’=triple, estado (char) que inicialmente es Libre (L), N° Total Servicios (int) inicialmente igual a cero, Ingreso total por Habitación (double) inicialmente igual a cero. Una habitación simple cuesta 100 soles, una habitación doble cuesta 150 soles y una habitación triple cuesta 250 soles.*

*- Reportar todos los datos (se realiza solo si hay datos ingresados).*

*- Agregar datos (se puede hacer siempre que haya espacio en los arreglos)*

*- Modificar datos por IdHabitación (se realiza solo si hay datos ingresados)*

*- Eliminar datos Nombre Habitación (se realiza solo si hay datos ingresados). Solo se puede eliminar si su estado es Libre (L). Si se elimina un IdHabitación, se elimina todos sus datos.*

*- Consultar por nombre de habitación (se realiza solo si hay datos ingresados).*

*- Consultar por IdHabitación (se realiza solo si hay datos ingresados)*

*- Ordenar los datos alfabéticamente en forma ascendente por nombre de habitación (se realiza solo si hay datos ingresados)*

*- Ordenar en forma descendente por IdHabitación (se realiza solo si hay datos ingresados)*

*- Reserva de habitación (se realiza solo si hay datos ingresados). Cuando se reserva una habitación, primero se debe preguntar qué tipo de habitación se requiere, y fecha; en base a la respuesta se muestran todas las habitaciones disponibles, luego se pide que se selecciones la habitación deseada y se cambia su estado a Ocupada. Luego se debe ingresar Nombre Cliente (string), Ciudad Origen (String), costo habitación (double), fecha e IdHabitación.*

*- Fin de servicio se hace en base al IdHabitación, significa que un servicio a terminado por lo tanto el estado de la habitación cambia a Libre Luego reportar todos los NT servicios realizados, indicando nombres de clientes. Ciudad de origen con el Id habitación y monto pagado. Además el monto total recaudado.*

*- Terminar el programa.*

*\*/*

import java.util.InputMismatchException; *// Para manejar excepciones de tipo InputMismatchException*

import java.util.Scanner; *// Para leer datos de entrada del usuario*

public class App {

  static Scanner input = new **Scanner**(System.in);

  static int max = 100;

  static int[] idHabitacion = new int[max];

  static String[] nombreHabitacion = new String[max];

  static char[] tipoHabitacion = new char[max];

  static char[] estado = new char[max];

  static int[] nTotalServicios = new int[max];

  static double[] montoHabitacion = new double[max];

  static String[] nombreCliente = new String[max];

  static String[] ciudadCliente = new String[max];

  static double[] costoHabitacion = new double[max];

  static String[] fechaReservaHabitacion = new String[max];

  static int cantidadHabitaciones = 0;

  static double costoFinal = 0;

  static boolean datosIngresados = false;

  static String[] tiposServicios = { "Desayuno", "Almuerzo", "Cena", "Lavandería", "Bar", "Cochera", "Spa", "Gimnasio",

      "Piscina", "Sauna" };

  static int[] serviciosSeleccionados = new int[10];

  static int[][] serviciosPorHabitacion = new int[max][10];

*/\*\**

*\* Método principal que ejecuta el sistema de gestión de hotel. Muestra el menú*

*\* principal y ejecuta la opción seleccionada por el usuario hasta que se*

*\* abandone*

*\* el programa. Las opciones del menú incluyen ingresar datos de habitaciones,*

*\* reportar todos los datos, agregar datos, modificar datos por IdHabitación,*

*\* eliminar datos por nombre de habitación, consultar por nombre de habitación,*

*\* consultar por IdHabitación, ordenar alfabéticamente por nombre de habitación,*

*\* ordenar descendentemente por IdHabitación, reservar habitación, fin de*

*\* servicio*

*\* y abandonar el programa.*

*\**

*\* @param args Argumentos de línea de comandos (no se utilizan)*

*\* @throws Exception Si ocurre un error al leer los datos de entrada del usuario*

*\*                   o al escribir en la salida estándar*

*\* @return void*

*\*/*

  public static void **main**(String[] *args*) throws Exception {

    boolean salir = false;

    while (!salir) {

      int opcion = **menu**();

      switch (opcion) {

        case 1:

          if (!datosIngresados) {

            cantidadHabitaciones = **ingresarDatos**(cantidadHabitaciones);

          } else {

            System.out.**println**("\nLos datos ya fueron ingresados");

          }

          break;

        case 2:

**reportarDatos**();

          break;

        case 3:

**agregarDatos**();

          break;

        case 4:

**modificarDatosPorId**();

          break;

        case 5:

**eliminarDatos**();

          break;

        case 6:

**consultarPorNombreHabitacion**();

          break;

        case 7:

**consultarPorIdHabitacion**();

          break;

        case 8:

**ordenarAlfabeticamentePorNombreHabitacion**();

          break;

        case 9:

**ordenarDescendentementePorIdHabitacion**();

          break;

        case 10:

**reservarHabitacion**();

          break;

        case 11:

**finDeServicio**();

          break;

        case 12:

          salir = **abandonarPrograma**(salir);

          break;

        default:

          System.out.**println**("\nOpción inválida");

          break;

      }

    }

  }

*/\*\**

*\* Muestra el menú principal del sistema de gestión de hotel y devuelve la*

*\* opción seleccionada.*

*\**

*\* @return La opción seleccionada como un número entero*

*\*/*

  public static int **menu**() {

    int opcion = 0;

**separador**("Hotel UPAO");

    System.out.**print**("1. Ingresar datos de N habitaciones de hotel\n");

    System.out.**print**("2. Reportar todos los datos\n");

    System.out.**print**("3. Agregar datos\n");

    System.out.**print**("4. Modificar datos por IdHabitación\n");

    System.out.**print**("5. Eliminar datos Nombre Habitación\n");

    System.out.**print**("6. Consultar por nombre de habitación\n");

    System.out.**print**("7. Consultar por IdHabitación\n");

    System.out.**print**("8. Ordenar los datos alfabéticamente en forma ascendente por nombre de habitación\n");

    System.out.**print**("9. Ordenar en forma descendente por IdHabitación\n");

    System.out.**print**("10. Reserva de habitación\n");

    System.out.**print**("11. Fin de servicio\n");

    System.out.**print**("12. Terminar el programa\n");

    System.out.**print**("Ingrese una opción: ");

    opcion = **validarEnteroPositivoEntre**(1, 12, "Ingrese una opción: ");

    return opcion;

  }

*/\*\**

*\* Limpia la pantalla de la consola.*

*\*/*

  public static void **limpiarPantalla**() {

    System.out.**print**("\033[H\033[2J");

    System.out.**flush**();

  }

*/\*\**

*\* Muestra un mensaje en la consola dependiendo de si se cumplió o no una*

*\* condición.*

*\**

*\* @param condicion La condición a evaluar.*

*\*/*

  public static void **mensaje**(boolean *condicion*) {

    if (*condicion* == true) {

      System.out.**println**("\nDatos ingresados correctamente");

    } else {

      System.out.**println**("\nNo hay datos ingresados");

    }

  }

*/\*\**

*\* Solicita al usuario que ingrese la cantidad de habitaciones y valida que sea*

*\* un número entero positivo entre 1 y 100.*

*\**

*\* @return La cantidad de habitaciones ingresada por el usuario como un número*

*\*         entero.*

*\*/*

  public static int **leerCantidadDeHabitaciones**() {

    int cantidadHabitaciones = 0;

    System.out.**print**("Ingrese la cantidad de habitaciones: ");

    cantidadHabitaciones = **validarEnteroPositivoEntre**(1, 100, "Ingrese la cantidad de habitaciones: ");

    return cantidadHabitaciones;

  }

*/\*\**

*\* Valida que el valor ingresado por el usuario sea un número entero positivo*

*\* entre un valor mínimo y máximo especificado. Si el valor ingresado no es*

*\* válido, se solicita al usuario que ingrese un nuevo valor.*

*\**

*\* @param min     El valor mínimo permitido.*

*\* @param max     El valor máximo permitido.*

*\* @param message El mensaje a mostrar al usuario para solicitar el valor.*

*\* @return El valor ingresado por el usuario como un número entero.*

*\*/*

  public static int **validarEnteroPositivoEntre**(int *min*, int *max*, String *message*) {

    int valor = 0;

    boolean esValido = false;

    do {

      try {

        valor = input.**nextInt**();

        if (valor >= *min* && valor <= *max*) {

          esValido = true;

        } else {

          System.out.**printf**("El valor debe estar entre %d y %d%n", *min*, *max*);

          System.out.**print**(*message*);

        }

      } catch (InputMismatchException *e*) {

        System.out.**println**("El valor debe ser un número entero");

        System.out.**print**(*message*);

        input.**nextLine**();

      }

    } while (!esValido);

    return valor;

  }

*/\*\**

*\* Solicita al usuario que ingrese los datos de N habitaciones de hotel y los*

*\* valida.*

*\**

*\* @param cantidadHabitaciones La cantidad de habitaciones a ingresar.*

*\* @return La cantidad de habitaciones ingresadas.*

*\*/*

  public static int **ingresarDatos**(int *cantidadHabitaciones*) {

**limpiarPantalla**();

    if (datosIngresados) {

      System.out.**println**("\nLos datos ya fueron ingresados");

    } else {

*cantidadHabitaciones* = **leerCantidadDeHabitaciones**();

      for (int i = 0; i < *cantidadHabitaciones*; i++) {

**ingresarDatosHabitacion**(i);

      }

      datosIngresados = true;

**mensaje**(datosIngresados);

    }

    return *cantidadHabitaciones*;

  }

*/\*\**

*\* Solicita al usuario que ingrese los datos de una habitación de hotel y los*

*\* valida.*

*\**

*\* @param index El índice de la habitación a ingresar.*

*\*/*

  public static void **ingresarDatosHabitacion**(int *index*) {

    System.out.**printf**("%nHabitación %d%n", *index* + 1);

    System.out.**print**("Ingrese el Id de la habitación (2 dígitos): ");

    idHabitacion[*index*] = **validarEnteroPositivoEntre**(10, 99, "Ingrese el Id de la habitación (2 dígitos): ");

**validarIdHabitacionUnico**(*index*);

    System.out.**print**("Ingrese el nombre de la habitación: ");

    nombreHabitacion[*index*] = **validarNombreHabitacion**("Ingrese el nombre de la habitación: ");

**validarNombreHabitacionUnico**(*index*);

    System.out.**print**("Ingrese el tipo de habitación (S = simple, D = doble, T = triple): ");

    tipoHabitacion[*index*] = **validarTipoHabitacion**(

        "Ingrese el tipo de habitación (S = simple, D = doble, T = triple): ");

    System.out.**print**("Ingrese el número total de servicios: ");

    nTotalServicios[*index*] = **validarEnteroPositivoEntre**(0, 10, "Ingrese el número total de servicios: ");

    if (nTotalServicios[*index*] > 0) {

**seleccionarServicios**(*index*);

    }

**calcularMontoHabitacion**(*index*);

    estado[*index*] = 'L';

  }

*/\*\**

*\* Verifica que el ID de la habitación sea único entre todas las habitaciones*

*\* ingresadas anteriormente. Si el ID no es único, solicita al usuario que*

*\* ingrese un nuevo ID hasta que se ingrese uno único*

*\**

*\* @param index El índice de la habitación que se está ingresando.*

*\*/*

  public static void **validarIdHabitacionUnico**(int *index*) {

    boolean idUnico = true;

    do {

      for (int j = 0; j < *index*; j++) {

        if (idHabitacion[*index*] == idHabitacion[j]) {

          System.out.**println**("El Id de la habitación debe ser único");

          System.out.**print**("Ingrese el Id de la habitación (2 dígitos): ");

          idHabitacion[*index*] = **validarEnteroPositivoEntre**(10, 99, "Ingrese el Id de la habitación (2 dígitos): ");

          while (idHabitacion[*index*] < 10 || idHabitacion[*index*] > 99) {

            System.out.**println**("El Id de la habitación debe ser un número de 2 dígitos");

            System.out.**print**("Ingrese el Id de la habitación (2 dígitos): ");

            idHabitacion[*index*] = **validarEnteroPositivoEntre**(10, 99, "Ingrese el Id de la habitación (2 dígitos): ");

          }

          idUnico = false;

          break;

        }

      }

      if (idUnico) {

        break;

      }

      idUnico = true;

    } while (true);

  }

*/\*\**

*\* Valida y devuelve un nombre de habitación ingresado por el usuario.*

*\* El nombre debe tener entre 2 y 30 caracteres de longitud.*

*\**

*\* @param prompt El mensaje para mostrar al usuario.*

*\* @return El nombre de la habitación validado.*

*\*/*

  public static String **validarNombreHabitacion**(String *prompt*) {

    String nombre = input.**next**().**toUpperCase**().**trim**();

    while (nombre.**length**() < 2 || nombre.**length**() > 30) {

      System.out.**println**("El nombre de la habitación debe tener entre 2 y 30 caracteres");

      System.out.**print**(*prompt*);

      nombre = input.**next**().**toUpperCase**().**trim**();

    }

    return nombre;

  }

*/\*\**

*\* Verifica que el nombre de la habitación sea único entre todas las*

*\* habitaciones ingresadas anteriormente. Si el nombre no es único, solicita al*

*\* usuario que ingrese un nuevo nombre hasta que se ingrese uno único.*

*\**

*\* @param index El índice de la habitación que se está ingresando.*

*\*/*

  public static void **validarNombreHabitacionUnico**(int *index*) {

    boolean nombreUnico = true;

    do {

      for (int j = 0; j < *index*; j++) {

        if (nombreHabitacion[*index*].**equals**(nombreHabitacion[j])) {

          System.out.**println**("El nombre de la habitación debe ser único");

          System.out.**print**("Ingrese el nombre de la habitación: ");

          nombreHabitacion[*index*] = **validarNombreHabitacion**("Ingrese el nombre de la habitación: ");

          nombreUnico = false;

          break;

        }

      }

      if (nombreUnico) {

        break;

      }

      nombreUnico = true;

    } while (true);

  }

*/\*\**

*\* Valida y devuelve un tipo de habitación ingresado por el usuario.*

*\* El tipo de habitación debe ser 'S', 'D' o 'T'.*

*\**

*\* @param prompt El mensaje para mostrar al usuario.*

*\* @return El tipo de habitación validado.*

*\*/*

  public static char **validarTipoHabitacion**(String *prompt*) {

    char tipo = input.**next**().**toUpperCase**().**charAt**(0);

    while (tipo != 'S' && tipo != 'D' && tipo != 'T') {

      System.out.**println**("El tipo de habitación debe ser S, D o T");

      System.out.**print**(*prompt*);

      tipo = input.**next**().**toUpperCase**().**charAt**(0);

    }

    return tipo;

  }

*/\*\**

*\* Permite al usuario seleccionar los servicios para una habitación en*

*\* particular.*

*\* Muestra los tipos de servicios disponibles y solicita al usuario que ingrese*

*\* el tipo de servicio deseado.*

*\* Verifica que el servicio ingresado sea único para la habitación y lo almacena*

*\* en la matriz de servicios por habitación.*

*\* Finalmente, muestra los servicios seleccionados para la habitación.*

*\**

*\* @param index El índice de la habitación para la cual se están seleccionando*

*\*              los servicios.*

*\*/*

  public static void **seleccionarServicios**(int *index*) {

    System.out.**println**("\nTipos de servicios");

    for (int j = 0; j < tiposServicios.length; j++) {

      System.out.**println**((j + 1) + ". " + tiposServicios[j]);

    }

    for (int j = 0; j < nTotalServicios[*index*]; j++) {

      System.out.**print**("Ingrese el tipo de servicio " + (j + 1) + ": ");

      serviciosSeleccionados[j] = **validarEnteroPositivoEntre**(1, 10, "Ingrese el tipo de servicio " + (j + 1) + ": ");

**validarServicioUnico**(*index*, j);

      serviciosPorHabitacion[*index*][j] = serviciosSeleccionados[j];

    }

    System.out.**println**("\nServicios seleccionados para la habitación");

    for (int j = 0; j < nTotalServicios[*index*]; j++) {

      System.out.**println**((j + 1) + ". " + tiposServicios[serviciosSeleccionados[j] - 1]);

    }

  }

*/\*\**

*\* Verifica que el servicio ingresado sea único para la habitación.*

*\* Si el servicio no es único, solicita al usuario que ingrese un servicio*

*\* diferente.*

*\**

*\* @param index El índice de la habitación para la cual se están seleccionando*

*\*              los servicios.*

*\* @param j     El índice del servicio que se está validando.*

*\*/*

  public static void **validarServicioUnico**(int *index*, int *j*) {

    boolean servicioUnico = true;

    do {

      for (int k = 0; k < *j*; k++) {

        if (serviciosSeleccionados[*j*] == serviciosSeleccionados[k]) {

          System.out.**println**("El servicio debe ser único");

          System.out.**print**("Ingrese el tipo de servicio " + (*j* + 1) + ": ");

          serviciosSeleccionados[*j*] = **validarEnteroPositivoEntre**(1, 10,

              "Ingrese el tipo de servicio " + (*j* + 1) + ": ");

          servicioUnico = false;

          break;

        }

      }

      if (servicioUnico) {

        break;

      }

      servicioUnico = true;

    } while (true);

  }

*/\*\**

*\* Calcula el monto total de la habitación en base a su tipo y la cantidad de*

*\* servicios seleccionados.*

*\**

*\* @param index El índice de la habitación para la cual se está calculando el*

*\*              monto.*

*\*/*

  public static void **calcularMontoHabitacion**(int *index*) {

    switch (tipoHabitacion[*index*]) {

      case 'S':

        if (nTotalServicios[*index*] == 0) {

          montoHabitacion[*index*] = 100;

        } else {

          montoHabitacion[*index*] = 100 + (nTotalServicios[*index*] \* 50);

        }

        break;

      case 'D':

        if (nTotalServicios[*index*] == 0) {

          montoHabitacion[*index*] = 150;

        } else {

          montoHabitacion[*index*] = 150 + (nTotalServicios[*index*] \* 50);

        }

        break;

      case 'T':

        if (nTotalServicios[*index*] == 0) {

          montoHabitacion[*index*] = 250;

        } else {

          montoHabitacion[*index*] = 250 + (nTotalServicios[*index*] \* 50);

        }

        break;

    }

  }

*/\*\**

*\* Verifica si una cadena de caracteres no contiene números.*

*\**

*\* @param cadena La cadena de caracteres a validar.*

*\* @return true si la cadena no contiene números, false en caso contrario.*

*\*/*

  public static boolean **validarCadenaDeCaracteres**(String *cadena*) {

    for (int i = 0; i < *cadena*.**length**(); i++) {

      if (*cadena*.**charAt**(i) >= '0' && *cadena*.**charAt**(i) <= '9') {

        return false;

      }

    }

    return true;

  }

*/\*\**

*\* Muestra un reporte de todas las habitaciones ingresadas hasta el momento,*

*\* incluyendo su id, nombre, tipo, estado, número total de servicios, servicios*

*\* seleccionados (si los hay) y monto total de la habitación.*

*\* Si no se han ingresado habitaciones, muestra un mensaje indicando que no se*

*\* han ingresado datos.*

*\*/*

  public static void **reportarDatos**() {

**limpiarPantalla**();

    if (cantidadHabitaciones == 0) {

      System.out.**println**("\nNo se han ingresado datos");

    } else {

**separador**("Reporte de Habitaciones");

      for (int i = 0; i < cantidadHabitaciones; i++) {

        System.out.**print**("Habitación " + (i + 1));

        System.out.**print**("\n- Id de la habitación: " + idHabitacion[i]);

        System.out.**print**("\n- Nombre de la habitación: " + nombreHabitacion[i]);

        System.out.**print**("\n- Tipo de habitación: " + tipoHabitacion[i]);

        System.out.**print**("\n- Estado de la habitación: " + estado[i]);

        System.out.**print**("\n- Número total de servicios: " + nTotalServicios[i]);

        if (nTotalServicios[i] > 0) {

          System.out.**print**("\n- Servicios seleccionados: ");

          for (int j = 0; j < nTotalServicios[i]; j++) {

            String service = tiposServicios[serviciosPorHabitacion[i][j] - 1];

            System.out.**print**("\n  " + (j + 1) + ". " + service + " $50");

          }

        }

        System.out.**print**("\n- Monto total de la habitación: $" + montoHabitacion[i]);

        System.out.**println**("\n");

      }

**separador**("");

    }

  }

*/\*\**

*\* Agrega los datos de una nueva habitación al sistema, si aún no se han*

*\* ingresado*

*\* la cantidad máxima de habitaciones permitidas.*

*\**

*\* Si se ingresan los datos correctamente, se muestra un mensaje de confirmación*

*\* y se*

*\* genera un reporte de todas las habitaciones ingresadas hasta el momento.*

*\**

*\* Si se ha alcanzado la cantidad máxima de habitaciones permitidas, se muestra*

*\* un*

*\* mensaje indicando que no se pueden ingresar más datos.*

*\*/*

  public static void **agregarDatos**() {

    if (cantidadHabitaciones == 100) {

      System.out.**println**("\nNo se pueden ingresar más datos");

    } else {

**ingresarDatosHabitacion**(cantidadHabitaciones);

      cantidadHabitaciones++;

**mensaje**(true);

**reportarDatos**();

    }

  }

  public static void **modificarDatosPorId**() {

    if (cantidadHabitaciones == 0) {

**mensaje**(false);

    } else {

      System.out.**print**("\nIngrese el Id de la habitación a modificar: ");

      int id = **validarEnteroPositivoEntre**(1, 100, "Ingrese el Id de la habitación a modificar: ");

      int i = 0;

      while (i < cantidadHabitaciones && id != idHabitacion[i]) {

        i++;

      }

      if (i == cantidadHabitaciones) {

        System.out.**println**("\nEl Id de la habitación no existe");

      } else {

        System.out.**println**("\nDatos de la habitación a modificar");

        System.out.**println**("Id de la habitación: " + idHabitacion[i]);

        System.out.**println**("Nombre de la habitación: " + nombreHabitacion[i]);

        System.out.**println**("Tipo de habitación: " + tipoHabitacion[i]);

        System.out.**println**("Estado de la habitación: " + estado[i]);

        System.out.**println**("Número total de servicios: " + nTotalServicios[i]);

        if (nTotalServicios[i] > 0) {

          System.out.**println**("Servicios seleccionados: ");

          for (int j = 0; j < nTotalServicios[i]; j++) {

            System.out.**println**((j + 1) + ". " + tiposServicios[serviciosPorHabitacion[i][j] - 1]);

          }

        }

        System.out.**println**("Costo de la habitación: " + montoHabitacion[i]);

        System.out.**println**("\nIngrese los nuevos datos de la habitación");

        System.out.**print**("Ingrese el Nombre de la habitación: ");

        nombreHabitacion[i] = **validarNombreHabitacion**("Ingrese el Nombre de la habitación: ");

*// validar que el nombre de la habitación no se repita*

        boolean nombreUnico = true;

        for (int j = 0; j < cantidadHabitaciones; j++) {

          if (nombreHabitacion[i].**equals**(nombreHabitacion[j]) && i != j) {

            nombreUnico = false;

            break;

          }

        }

        while (!nombreUnico) {

          System.out.**println**("\nEl Nombre de la habitación ya existe");

          System.out.**print**("Ingrese el Nombre de la habitación: ");

          nombreHabitacion[i] = **validarNombreHabitacion**("Ingrese el Nombre de la habitación: ");

          nombreUnico = true;

          for (int j = 0; j < cantidadHabitaciones; j++) {

            if (nombreHabitacion[i].**equals**(nombreHabitacion[j]) && i != j) {

              nombreUnico = false;

              break;

            }

          }

        }

        System.out.**print**("\nIngrese el Tipo de habitación (S = Simple, D = Doble, T = Triple): ");

        tipoHabitacion[i] = **validarTipoHabitacion**("Ingrese el Tipo de habitación: ");

        System.out.**print**("\nIngrese el Número total de servicios: ");

        nTotalServicios[i] = **validarEnteroPositivoEntre**(0, 10, "Ingrese el Número total de servicios: ");

        if (nTotalServicios[i] > 0) {

*// mostrar los tipos de servicios*

          System.out.**println**("\nTipos de servicios");

          for (int j = 0; j < tiposServicios.length; j++) {

            System.out.**println**((j + 1) + ". " + tiposServicios[j]);

          }

          System.out.**println**("\nIngrese los servicios seleccionados");

          for (int j = 0; j < nTotalServicios[i]; j++) {

            System.out.**print**("Ingrese el servicio " + (j + 1) + ": ");

            serviciosPorHabitacion[i][j] = **validarEnteroPositivoEntre**(1, 10,

                "Ingrese el servicio " + (j + 1) + ": ");

          }

        }

**calcularMontoHabitacion**(i);

**mensaje**(true);

**reportarDatos**();

      }

    }

  }

  public static void **eliminarDatos**() {

    if (cantidadHabitaciones == 0) {

**mensaje**(false);

    } else {

      System.out.**print**("\nIngrese el Id de la habitación a eliminar: ");

      int id = **validarEnteroPositivoEntre**(1, 100, "Ingrese el Id de la habitación a eliminar: ");

      int i = 0;

      while (i < cantidadHabitaciones && id != idHabitacion[i]) {

        i++;

      }

      if (i == cantidadHabitaciones) {

        System.out.**println**("\nEl Id de la habitación no existe");

      } else {

        System.out.**print**("\n¿Está seguro que desea eliminar la habitación? (S/N): ");

        String respuesta = input.**next**().**toUpperCase**().**trim**();

        while (!respuesta.**equals**("S") && !respuesta.**equals**("N")) {

          System.out.**println**("Respuesta inválida. Por favor ingrese S o N.");

          System.out.**print**("\n¿Está seguro que desea eliminar la habitación? (S/N): ");

          respuesta = input.**next**().**toUpperCase**().**trim**();

        }

        if (respuesta.**equals**("S")) {

          for (int j = i; j < cantidadHabitaciones - 1; j++) {

            idHabitacion[j] = idHabitacion[j + 1];

            nombreHabitacion[j] = nombreHabitacion[j + 1];

            tipoHabitacion[j] = tipoHabitacion[j + 1];

            estado[j] = estado[j + 1];

            nTotalServicios[j] = nTotalServicios[j + 1];

            montoHabitacion[j] = montoHabitacion[j + 1];

            for (int k = 0; k < nTotalServicios[j + 1]; k++) {

              serviciosPorHabitacion[j][k] = serviciosPorHabitacion[j + 1][k];

            }

          }

          cantidadHabitaciones--;

**mensaje**(true);

**reportarDatos**();

        } else {

          System.out.**println**("\nLa habitación no ha sido eliminada.");

**reportarDatos**();

        }

      }

    }

  }

  public static void **consultarPorNombreHabitacion**() {

    if (cantidadHabitaciones == 0) {

**mensaje**(false);

    } else {

      System.out.**print**("\nIngrese el Nombre de la habitación a consultar: ");

      String nombre = **validarNombreHabitacion**("Ingrese el Nombre de la habitación a consultar: ");

      int i = 0;

      while (i < cantidadHabitaciones && !nombre.**equals**(nombreHabitacion[i])) {

        i++;

      }

      if (i == cantidadHabitaciones) {

**limpiarPantalla**();

        System.out.**println**("\nEl Nombre de la habitación no existe");

      } else {

        System.out.**println**("\n- Id de la habitación: " + idHabitacion[i]);

        System.out.**println**("- Nombre de la habitación: " + nombreHabitacion[i]);

        System.out.**println**("- Tipo de habitación: " + tipoHabitacion[i]);

        System.out.**println**("- Estado: " + estado[i]);

        if (nTotalServicios[i] > 0) {

          System.out.**println**("Servicios seleccionados: ");

          for (int j = 0; j < nTotalServicios[i]; j++) {

            System.out.**println**((j + 1) + ". " + tiposServicios[serviciosPorHabitacion[i][j] - 1]);

          }

        }

        System.out.**println**("Costo de la habitación: " + montoHabitacion[i]);

      }

    }

  }

  public static void **consultarPorIdHabitacion**() {

    if (cantidadHabitaciones == 0) {

**mensaje**(false);

    } else {

      System.out.**print**("\nIngrese el Id de la habitación a consultar: ");

      int id = **validarEnteroPositivoEntre**(1, 100, "Ingrese el Id de la habitación a consultar: ");

      int i = 0;

      while (i < cantidadHabitaciones && id != idHabitacion[i]) {

        i++;

      }

      if (i == cantidadHabitaciones) {

        System.out.**println**("\nEl Id de la habitación no existe");

      } else {

        System.out.**println**("\n- Id de la habitación: " + idHabitacion[i]);

        System.out.**println**("- Nombre de la habitación: " + nombreHabitacion[i]);

        System.out.**println**("- Tipo de habitación: " + tipoHabitacion[i]);

        System.out.**println**("- Estado: " + estado[i]);

        if (nTotalServicios[i] > 0) {

          System.out.**println**("Servicios seleccionados: ");

          for (int j = 0; j < nTotalServicios[i]; j++) {

            System.out.**println**((j + 1) + ". " + tiposServicios[serviciosPorHabitacion[i][j] - 1]);

          }

        }

        System.out.**println**("Costo de la habitación: " + montoHabitacion[i]);

      }

    }

  }

  public static void **ordenarAlfabeticamentePorNombreHabitacion**() {

**limpiarPantalla**();

    if (cantidadHabitaciones == 0) {

**mensaje**(false);

    } else {

      for (int i = 0; i < cantidadHabitaciones - 1; i++) {

        for (int j = 0; j < cantidadHabitaciones - i - 1; j++) {

          if (nombreHabitacion[j].**compareTo**(nombreHabitacion[j + 1]) > 0) {

            int auxId = idHabitacion[j];

            idHabitacion[j] = idHabitacion[j + 1];

            idHabitacion[j + 1] = auxId;

            String auxNombre = nombreHabitacion[j];

            nombreHabitacion[j] = nombreHabitacion[j + 1];

            nombreHabitacion[j + 1] = auxNombre;

            char auxTipo = tipoHabitacion[j];

            tipoHabitacion[j] = tipoHabitacion[j + 1];

            tipoHabitacion[j + 1] = auxTipo;

            char auxEstado = estado[j];

            estado[j] = estado[j + 1];

            estado[j + 1] = auxEstado;

            int auxNTotalServicios = nTotalServicios[j];

            nTotalServicios[j] = nTotalServicios[j + 1];

            nTotalServicios[j + 1] = auxNTotalServicios;

            double auxMontoHabitacion = montoHabitacion[j];

            montoHabitacion[j] = montoHabitacion[j + 1];

            montoHabitacion[j + 1] = auxMontoHabitacion;

            for (int k = 0; k < nTotalServicios[j]; k++) {

              int auxServiciosPorHabitacion = serviciosPorHabitacion[j][k];

              serviciosPorHabitacion[j][k] = serviciosPorHabitacion[j + 1][k];

              serviciosPorHabitacion[j + 1][k] = auxServiciosPorHabitacion;

            }

          }

        }

      }

**mensaje**(true);

**reportarDatos**();

    }

  }

  public static void **ordenarDescendentementePorIdHabitacion**() {

**limpiarPantalla**();

    if (cantidadHabitaciones == 0) {

**mensaje**(false);

    } else {

      for (int i = 0; i < cantidadHabitaciones - 1; i++) {

        for (int j = 0; j < cantidadHabitaciones - i - 1; j++) {

          if (idHabitacion[j] < idHabitacion[j + 1]) {

            int auxId = idHabitacion[j];

            idHabitacion[j] = idHabitacion[j + 1];

            idHabitacion[j + 1] = auxId;

            String auxNombre = nombreHabitacion[j];

            nombreHabitacion[j] = nombreHabitacion[j + 1];

            nombreHabitacion[j + 1] = auxNombre;

            char auxTipo = tipoHabitacion[j];

            tipoHabitacion[j] = tipoHabitacion[j + 1];

            tipoHabitacion[j + 1] = auxTipo;

            char auxEstado = estado[j];

            estado[j] = estado[j + 1];

            estado[j + 1] = auxEstado;

            int auxNTotalServicios = nTotalServicios[j];

            nTotalServicios[j] = nTotalServicios[j + 1];

            nTotalServicios[j + 1] = auxNTotalServicios;

            double auxMontoHabitacion = montoHabitacion[j];

            montoHabitacion[j] = montoHabitacion[j + 1];

            montoHabitacion[j + 1] = auxMontoHabitacion;

            for (int k = 0; k < nTotalServicios[j]; k++) {

              int auxServiciosPorHabitacion = serviciosPorHabitacion[j][k];

              serviciosPorHabitacion[j][k] = serviciosPorHabitacion[j + 1][k];

              serviciosPorHabitacion[j + 1][k] = auxServiciosPorHabitacion;

            }

          }

        }

      }

**mensaje**(true);

**reportarDatos**();

    }

  }

  public static void **reservarHabitacion**() {

    if (cantidadHabitaciones == 0) {

**mensaje**(false);

    } else {

      char tipoHabitacionDeseada;

**separador**("RESERVAR HABITACIÓN");

      System.out.**print**("Ingrese el tipo de habitación que desea reservar: ");

      tipoHabitacionDeseada = input.**next**().**toUpperCase**().**charAt**(0);

      while (tipoHabitacionDeseada != 'S' && tipoHabitacionDeseada != 'D' && tipoHabitacionDeseada != 'T') {

        System.out.**print**("\nTipo de habitación inválido. Ingrese nuevamente: ");

        tipoHabitacionDeseada = input.**next**().**toUpperCase**().**charAt**(0);

      }

      System.out.**print**("\nIngrese la fecha de reserva (dd/mm/yyyy): ");

      String fechaReserva = input.**next**();

      int habitacionesDisponibles = 0;

      for (int i = 0; i < cantidadHabitaciones; i++) {

        if (tipoHabitacion[i] == tipoHabitacionDeseada && estado[i] == 'L') {

          System.out.**println**("\nId de la habitación: " + idHabitacion[i]);

          System.out.**println**("Nombre de la habitación: " + nombreHabitacion[i]);

          System.out.**println**("Tipo de habitación: " + tipoHabitacion[i]);

          System.out.**println**("Estado de la habitación: " + estado[i]);

          System.out.**println**("Número total de servicios: " + nTotalServicios[i]);

          if (nTotalServicios[i] > 0) {

            System.out.**println**("Servicios de la habitación: ");

            for (int j = 0; j < nTotalServicios[i]; j++) {

              System.out.**println**("  " + serviciosPorHabitacion[i][j]);

            }

          }

          System.out.**println**("Costo de la habitación: " + montoHabitacion[i]);

          habitacionesDisponibles++;

        }

      }

      if (habitacionesDisponibles == 0) {

        System.out.**println**("\nNo hay habitaciones disponibles");

      } else {

        System.out.**print**("\nIngrese el Id de la habitación que desea reservar: ");

        int idHabitacionDeseada = input.**nextInt**();

        int i = 0;

        while (i < cantidadHabitaciones && idHabitacion[i] != idHabitacionDeseada) {

          i++;

        }

        if (i == cantidadHabitaciones) {

          System.out.**print**("\nEl Id de la habitación no existe");

        } else {

          if (estado[i] == 'O') {

            System.out.**print**("\nLa habitación ya está ocupada");

          } else {

            String cliente, clienteCiudad;

            estado[i] = 'O';

            fechaReservaHabitacion[i] = fechaReserva;

            costoHabitacion[i] = montoHabitacion[i];

            System.out.**print**("\nIngrese el nombre del cliente: ");

            cliente = input.**next**();

            while (!**validarCadenaDeCaracteres**(cliente)) {

              System.out.**print**("\nNombre del cliente inválido. Ingrese nuevamente: ");

              cliente = input.**next**();

            }

            nombreCliente[i] = cliente;

            System.out.**print**("\nIngrese la ciudad del cliente: ");

            clienteCiudad = input.**next**();

            while (!**validarCadenaDeCaracteres**(clienteCiudad)) {

              System.out.**print**("\nCiudad del cliente inválida. Ingrese nuevamente: ");

              clienteCiudad = input.**next**();

            }

            ciudadCliente[i] = clienteCiudad;

**reportarDatos**();

*// imprimir factura*

**separador**("Factura");

            System.out.**println**("Id de la habitación: " + idHabitacion[i]);

            System.out.**println**("Nombre de la habitación: " + nombreHabitacion[i]);

            System.out.**println**("Tipo de la habitación: " + tipoHabitacion[i]);

            System.out.**println**("Estado de la habitación: " + estado[i]);

            System.out.**println**("Número total de servicios: " + nTotalServicios[i]);

            System.out.**println**("Costo de la habitación: " + montoHabitacion[i]);

            System.out.**println**("Nombre del cliente: " + nombreCliente[i]);

            System.out.**println**("Ciudad de origen del cliente: " + ciudadCliente[i]);

            System.out.**println**("\nReserva realizada con éxito");

          }

        }

      }

    }

  }

  public static void **finDeServicio**() {

    if (cantidadHabitaciones == 0) {

**mensaje**(false);

    } else {

      System.out.**println**("Ingrese el Id de la habitación que desea dar fin de servicio: ");

      int idHabitacionDeseada = input.**nextInt**();

      int i = 0;

      while (i < cantidadHabitaciones && idHabitacion[i] != idHabitacionDeseada) {

        i++;

      }

      if (i == cantidadHabitaciones) {

        System.out.**println**("\nEl Id de la habitación no existe");

      } else {

        if (estado[i] == 'L') {

          System.out.**println**("\nLa habitación no está ocupada");

        } else {

          estado[i] = 'L';

*// calculo del costo final solo de la habitación que se dio fin de servicio*

          for (int j = 0; j < cantidadHabitaciones; j++) {

            if (idHabitacion[j] == idHabitacionDeseada) {

              costoFinal += costoHabitacion[j];

            }

          }

**reportarDatos**();

**separador**("Fin de servicio");

          System.out.**println**("\nFin de servicio realizado con éxito");

          System.out.**println**("\nId de la habitación: " + idHabitacion[i]);

          System.out.**println**("Nombre de la habitación: " + nombreHabitacion[i]);

          System.out.**println**("Nombre del cliente: " + nombreCliente[i]);

          System.out.**println**("Ciudad de origen del cliente: " + ciudadCliente[i]);

          System.out.**println**("Cantidad de servicios: " + nTotalServicios[i]);

          System.out.**println**("Monto pagado: " + costoHabitacion[i]);

          System.out.**println**("Monto total recaudado: " + costoFinal);

*// se reinician los valores de la habitación que se dio fin de servicio*

          fechaReservaHabitacion[i] = "";

          nombreCliente[i] = "";

          ciudadCliente[i] = "";

        }

      }

    }

  }

  public static boolean **abandonarPrograma**(boolean *abandonar*) {

    System.out.**println**("\n¿Está seguro que desea salir del programa? (S/N)");

    String respuesta = input.**next**();

    if (respuesta.**equalsIgnoreCase**("S")) {

      System.out.**println**("\n¡Gracias por utilizar nuestro sistema de reservas de hotel!");

*abandonar* = true;

    } else if (respuesta.**equalsIgnoreCase**("N")) {

      System.out.**println**("\nContinuando en el programa...");

    } else {

      System.out.**println**("\nRespuesta inválida. Continuando en el programa...");

    }

    return *abandonar*;

  }

  public static void **separador**(String *mensaje*) {

    System.out.**print**("\n\*----------------- ");

    if (*mensaje* != "") {

      System.out.**print**(*mensaje*);

    }

    System.out.**println**(" -----------------\*");

  }

}