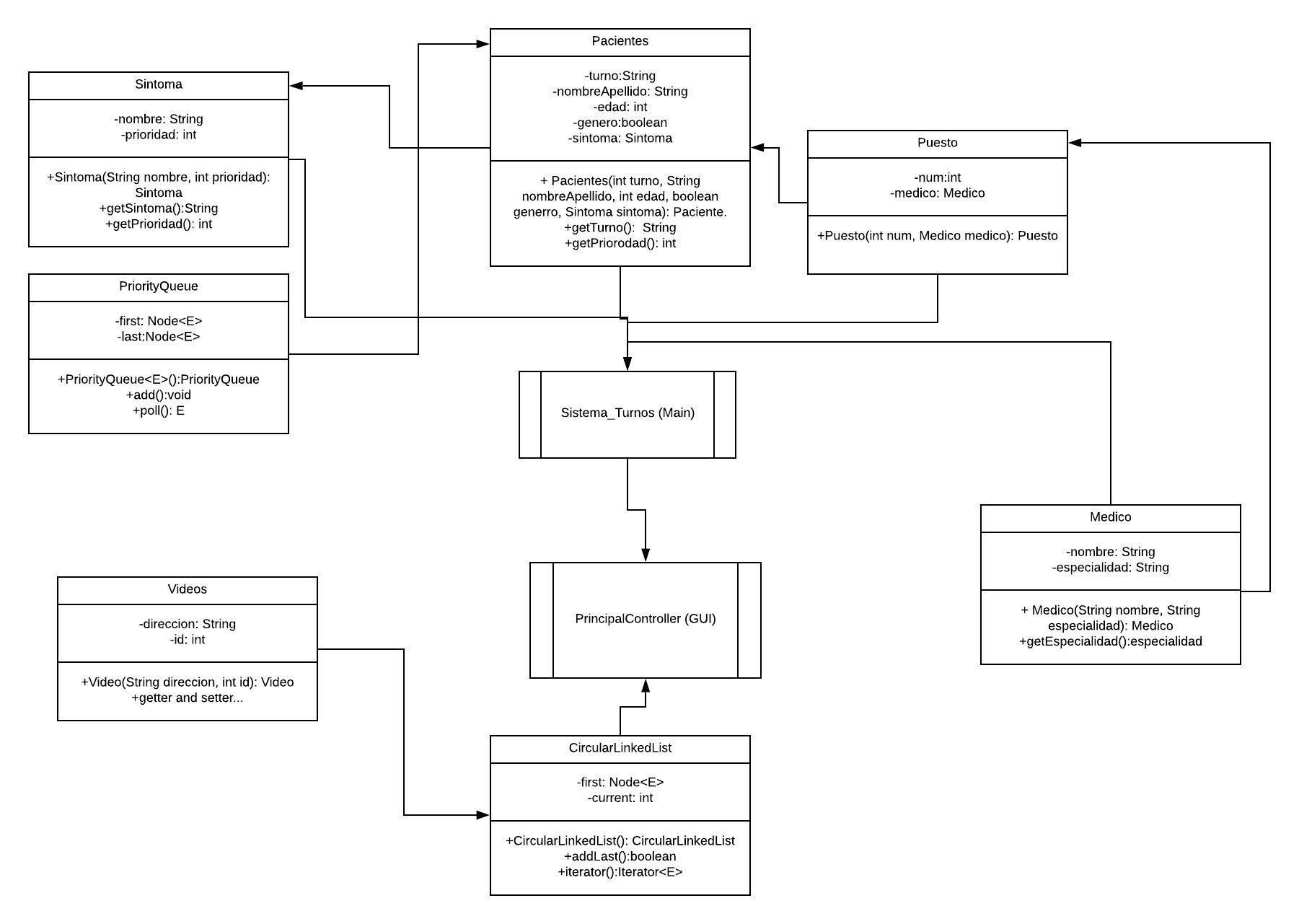
|  |  |
| --- | --- |
| **Estructuras de Datos** | |
| **Grupo N° 6** | **Tema:** Reporte de Proyecto |
| **Paralelo N° 2** | **Fecha:** lunes 20 de julio del 2020 |
| **Integrantes:** Mauricio Ortiz Lascano, Raúl Villao Rodríguez, Sebastián Benalcázar | |

**UML**



**CLASES IMPLEMENTADAS**

**Video**

Clase destinada para los videos que se reproducen en la ventana principal del programa. Su constructor recibe como parámetros el nombre y la duración del video en segundos.

**Medico**

Clase para los médicos de nuestra aplicación, en su constructor recibe el nombre, genero, cedula y especialidad.

**Paciente**

Clase para los pacientes que se van a atender, en su constructor recibe nombre, edad, genero, síntoma y turno.

**Síntoma**

Clase para los posibles síntomas que puede tener un paciente, en su constructor recibe el nombre y la gravedad del síntoma, la misma que se utilizara luego para ordenar a los pacientes siendo el más grave atendido primero.

**Puesto**

Clase para los puestos existentes dentro de nuestro programa, en su constructor recibe un número entero que lo identifica y un medico asociado al puesto que vamos a crear.

**TDA’S UTILIZADOS**

**SimplyLinkedList**

Utilizamos este TDA para crear las listas “síntomas”, “puestos”, “turnos” y “médicos”; las mismas que contendrá instancias de las clases Medico, Puesto, Turno y Síntoma. Decidimos usar este TDA en vez de ArrayList, ya que la información que contienen las listas será actualizada constantemente (se agregara o se eliminará información) por lo que se tendría que utilizar AddCapacity seguido y esto haría que nuestro programa sea menos eficiente.

**CircularLinkedList**

Utilizamos este TDA para poder crear una lista de videos que se pudiera repetir indefinidamente la misma que sería utilizada en la ventana que aparecerá al iniciar el programa.

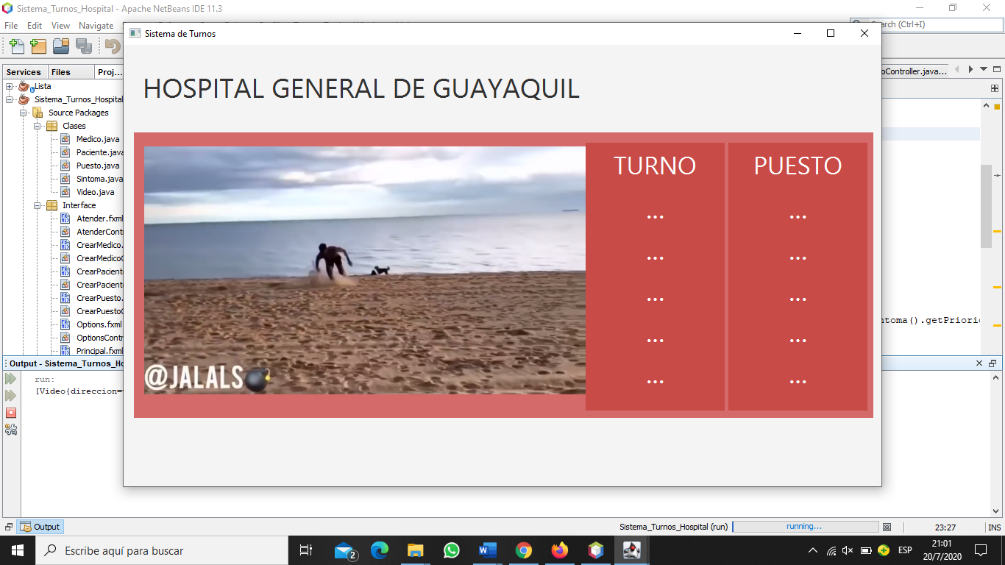
**PriorityQueue**

Utilizamos este TDA para crear la lista “pacientes”, la cual debe estar ordenada dependiendo de la gravedad de cada paciente.

**INTERFACE**

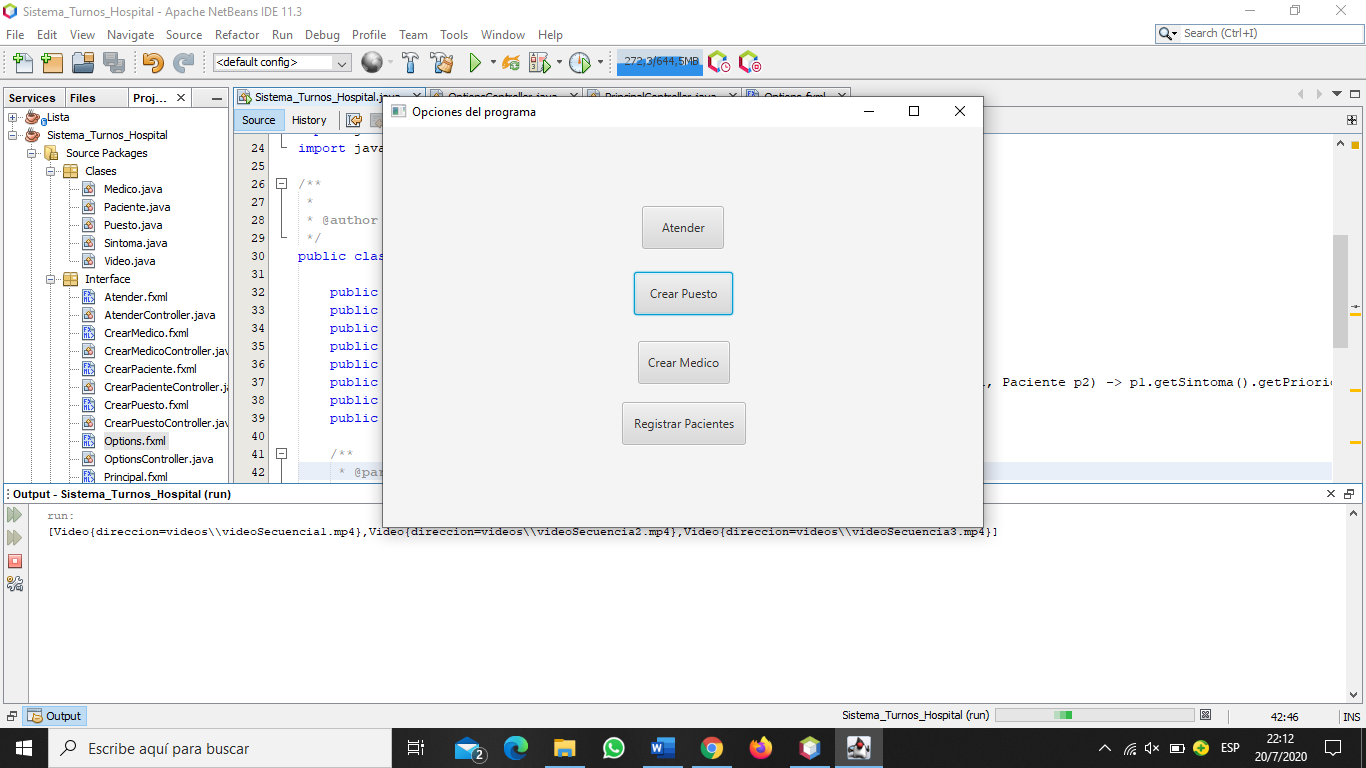
**Principal**

Esta es la ventana principal, la cual aparecerá al iniciar el programa, aquí se podrá visualizar el nombre del hospital, así como una pequeña tabla que mostrará cada turno con su respectivo puesto. En adición a esto, en la parte izquierda se mostrarán videos para que los usuarios se puedan entretener mientras esperan su turno.



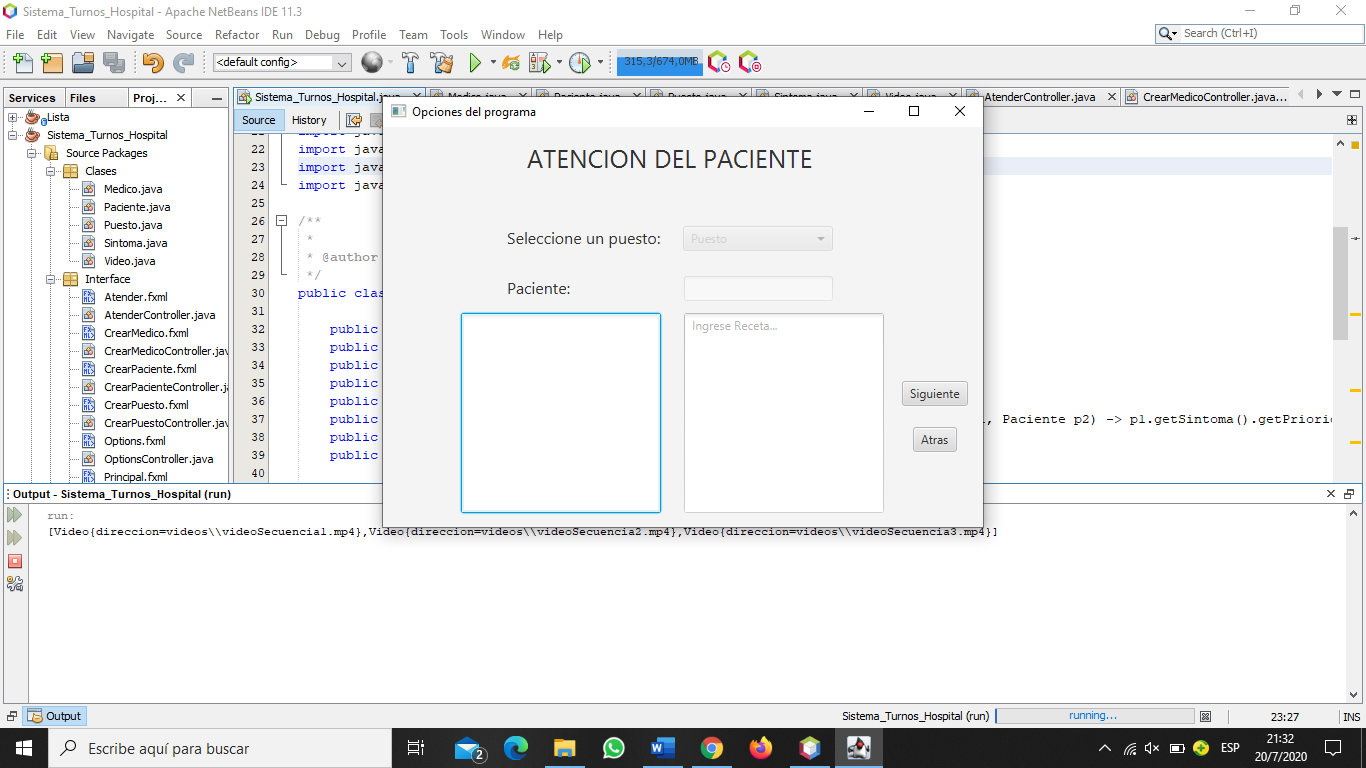
**Options**

Esta es la ventada desde la cual podremos acceder a todas las acciones que tiene el programa, aquí habrá 4 botones los cuales nos llevaran secciones diferentes de nuestro programa, tales como “Atender”, “Crear Puesto”, “Crear Medico” y “Registrar Pacientes”.

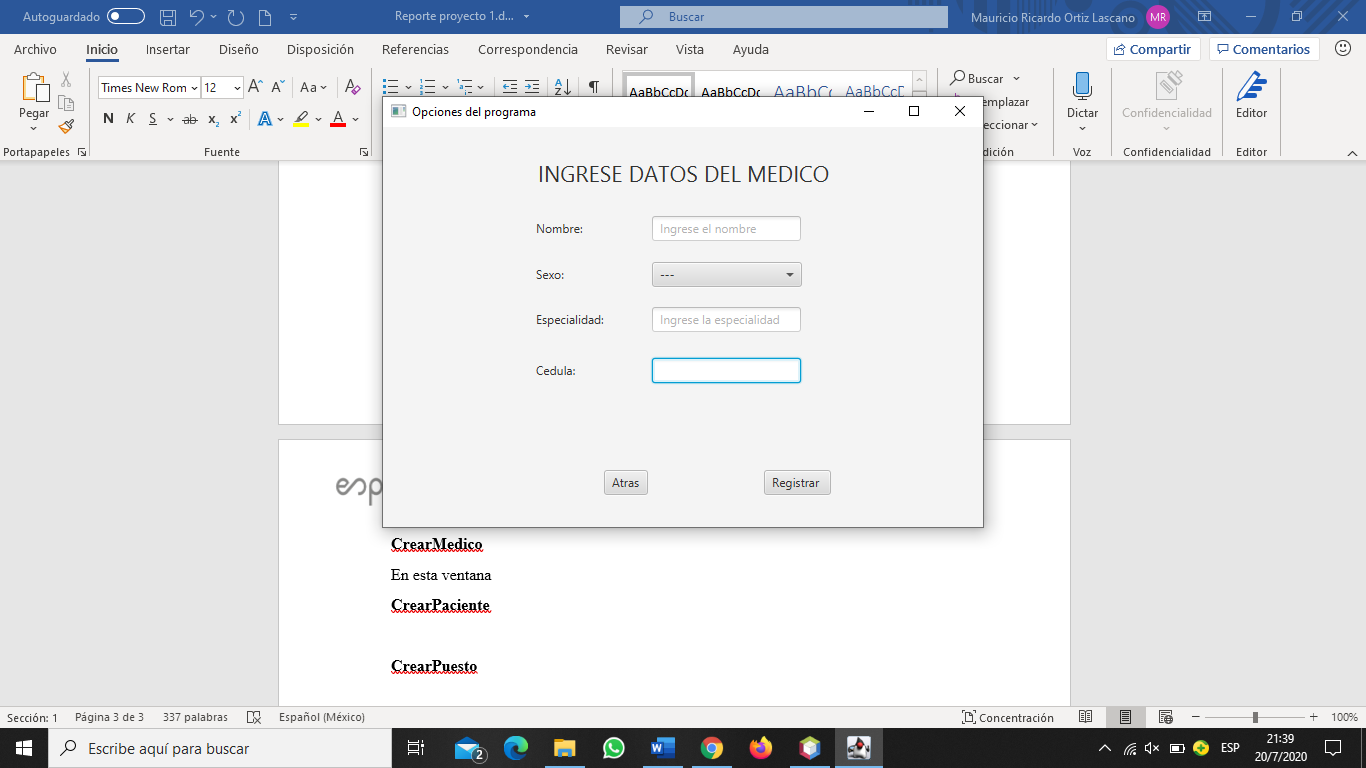


**Atender**

Esta ventana está dedicada a la atención de los pacientes, aquí se tendrá que seleccionar el puesto e ingresar el nombre del paciente al que atenderá, así como el diagnostico y la receta luego de la atención.

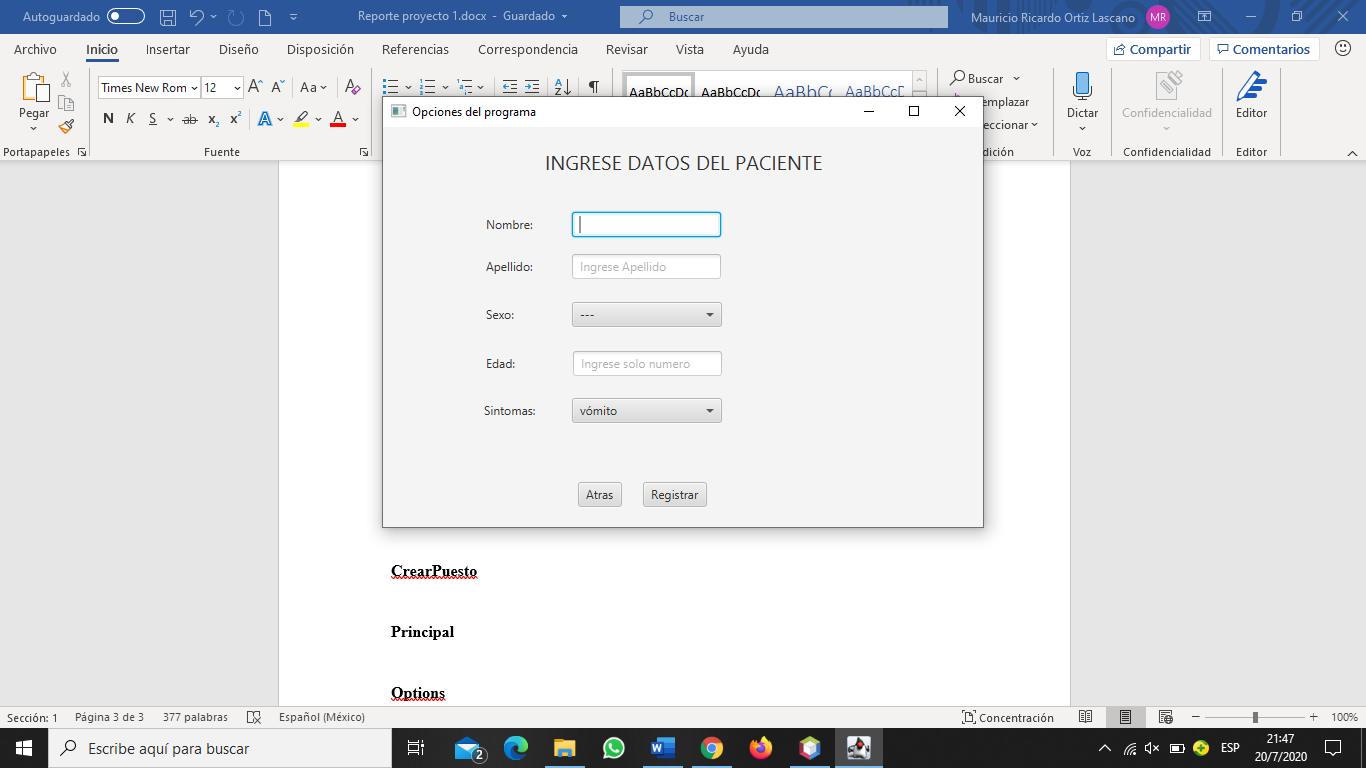


**CrearMedico**

En esta ventana se podrán ingresar a los médicos del hospital en el programa, aquí se tendrá que escribir datos como el nombre, la especialidad y la cedula del médico; además encontraremos un ComboBox donde el usuario deberá elegir el género del médico.

**CrearPaciente**

En esta ventana se podrán ingresar a los pacientes en el programa para posteriormente ser atendidos, aquí se tendrán que escribir los datos del paciente, tales como nombre, apellido y edad; además encontraremos dos ComboBox donde el usuario deberá elegir el género del paciente y el síntoma que el paciente presenta.



**CrearPuesto**

En esta ventana se crearán los puestos en los que se colocaran a los diferentes pacientes, aquí se tendrá que ingresar el ID del puesto (número) que identificara a cada puesto y será único, además a través de un ComboBox se deberá asociar el puesto creado con un médico en específico.

