|  |  |  |
| --- | --- | --- |
| **Nombre:**  Tania Lizbeth Torres Ávila | | **Matrícula:**  2928635 |
| **Nombre del curso:**  Computación en Java | **Nombre del profesor:**  Edgar Hernández Hernández | |
| **Módulo:**  Módulo 2 | **Actividad:**  Avance de Evidencia 2 | |
| **Fecha: 21** /04/2021 | | |
| **Bibliografía:** | | |

**Avance 2**

El participante aplicará sus conocimientos obtenidos en el curso para crear un programa que simulará un sistema de administración de citas para un consultorio clínico. El programa deberá poder realizar las siguientes acciones:

•Dar de alta doctores.

•Dar de alta pacientes

•Crear una cita con fecha y hora

.•Relacionar una cita con un doctor y un paciente.

Tener control de acceso mediante administradores, esto es, solo ciertos usuarios podrán acceder al sistema mediante un identificador y una contraseña.

**Dar de alta doctores:**

se deberán poder dar de alta los doctores del consultorio médico, los datos básicos serán:

Identificador único.•

Nombre completo.

Especialidad.

**Dar de alta pacientes:**

se deberán poder registrar los pacientes que acudan al consultorio médico, los datos básicos serán

Identificador único.

•Nombre completo.

**Crear una cita con fecha y hora**:

se deberán poder crear múltiples citas, los datos básicos serán:•

Identificador único

.•Fecha y hora de la cita

.•Motivo de la cita.

•Relacionar una cita con un doctor y un paciente:

cada una de las citas creadas deberá estar relacionada con un doctor y un paciente.

•Tener control de acceso mediante administradores: solo ciertos usuarios podrán acceder al sistema mediante un identificador y una contraseña

.•Toda la información deberá ser almacenada en archivos de texto plano con formato CSV, o en su defecto utilizar algún formato más avanzado como JSON o XML

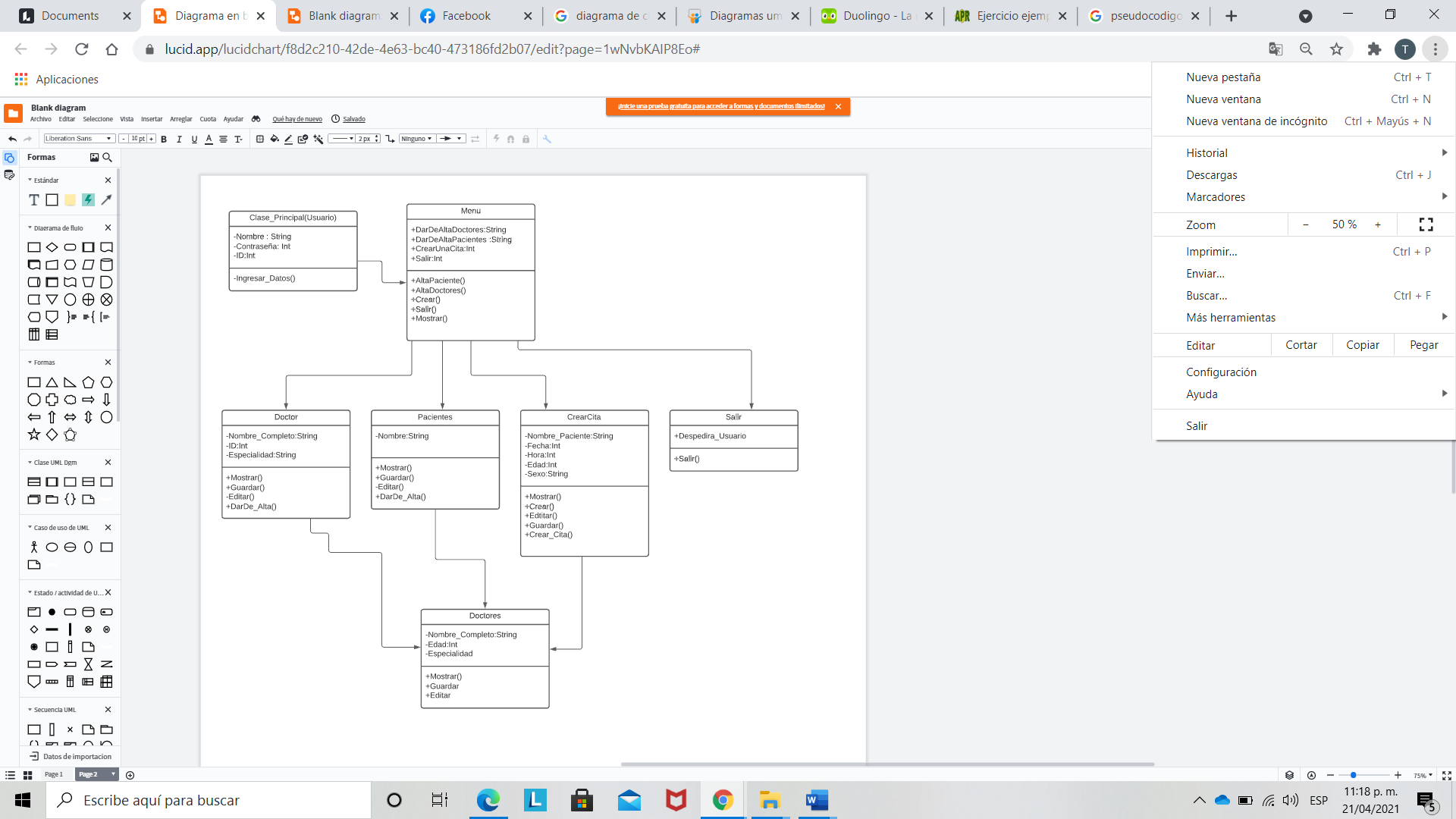
**Diseño del programa (diagrama de clases)**

Después de avanzar en los conocimientos sobre programación orientada a objetos, se realizará un diagrama de clases donde se desglosarán los componentes de la aplicación.

Se deberán estructurar los componentes de acuerdo con las funcionalidades del sistema, por ejemplo:

* Clase Principal
* Clase para Doctor
* Clase para Paciente
* Clase para cita

El reto es pensar en otras clases, tanto abstractas como concretas, así como posibles interfaces que sean necesarias para implementar correctamente las funcionalidades del programa, no olvidar tomar en cuenta los conceptos de herencia y polimorfismo.



**2.Pseudocodigo**

Con base en el diagrama parte de la entrega 1, traducirlo a pseudocodigo .Si es necesario , mejorar el diagrama de flujo.

**Programa Sistema\_AdministracionDeCitas**

**Variables**

Int :Opción

String: nombreDoctor

String :nombrePaciente

Int :IDdoctor

Int: fecha

Int :hora

String: Especialidad

String:MotivoCita

Int :Contraseña

**Inicio**

Bienvenida al usuario

Pedir contraseña

Pedir ID

Leer Datos

Mostrar Menu

1.DarDe\_AltaDoctores

2.Dar\_De\_AltaPacientes

3.Crear\_Cita

4Salir

Si Opción == 1

Entonces

Dar de alta Doctores

**Variables**

Int :Identificador único

String :Nombre completo.

String: Especialidad.

Leer Datos

Guardar Datos

Fin\_Para

Mostrar Datos

Si Opción ==2

Dar de alta Pacientes

**Variables**

Identificador único.

Nombre completo.

Leer Datos

Guardar Datos

Fin\_Para

Mostrar Datos

Si Opción==3

Crear Cita

**Variables**

Int :Hora

Int :Fecha

String :Nombre

Leer Datos

Guardar Datos

Fin\_Para

Mostrar Datos

Si Opción ==4

Salir del programa

**Fin**

**Conclusión:**

Al realizar esta actividad fue difícil e investigue,

pero logre realizarlo