**TECNICATURA SUPERIOR**

**EN**

**PROGRAMACIÓN**

**TRABAJO PRÁCTICO**

**DIAGRAMA UML**

**Cátedra:** Gestión de Desarrollo de Software.

**Nivel:** 2° año.

**Año lectivo:** 2024-25

**Grupo:** Solución 202

**Integrantes y matrículas individuales:**

* Gauna, Gimena – 11091
* Herrera, Emmanuel – 11108
* Montenegro, Maximiliano – 11160
* Orellana, Juan – 11182
* Poblete, Franco - 11196
* Velázquez, Ignacio – 11271

## **Caso 2: Clínica Médica**

## **1. Introducción**

El presente trabajo tiene como objetivo analizar y representar mediante diagramas UML el funcionamiento de un sistema de gestión para una clínica médica.  
 El sistema debe permitir la administración de turnos, consultas médicas, estudios y procesos de facturación.  
 Para ello, se elaborarán dos diagramas principales: **diagrama de casos de uso** y **diagrama de clases**, que reflejan la estructura y comportamiento general del sistema.

## **2. Análisis de requerimientos**

### **Actores identificados**

* **Paciente:** solicita turnos, recibe órdenes de estudio y consulta resultados.
* **Recepcionista:** confirma la asistencia del paciente y abre su historia clínica.
* **Médico:** registra diagnósticos, emite recetas electrónicas y solicita estudios.
* **Técnico de laboratorio:** carga resultados de estudios en el sistema.
* **Área administrativa:** factura las consultas al seguro y registra los pagos.
* **Sistema:** verifica disponibilidad, genera órdenes y asocia resultados.

## **3. Casos de uso identificados**

### **Paciente**

1. **Solicitar turno**  
    *Incluye:* Verificar disponibilidad y Confirmar cita.
2. **Recibir orden de estudio.**
3. **Consultar resultados de estudios.**

### **Recepcionista**

1. **Confirmar asistencia de paciente.**
2. **Abrir historia clínica.**

### **Médico**

1. **Registrar diagnóstico.**
2. **Emitir receta electrónica.**
3. **Solicitar estudio.**  
    *Incluye:* Generar orden de estudio.

### **Técnico de laboratorio**

1. **Cargar resultados de estudio.**  
    *Incluye:* Asociar resultado a historia clínica.

### **Área administrativa**

1. **Facturar consulta al seguro.**
2. **Registrar pago.**  
    *Extiende:* Facturar consulta.

## **4. Descripción breve de los principales casos de uso**

* **CU1 – Solicitar turno:** El paciente selecciona especialidad, médico, día y hora. El sistema verifica disponibilidad y confirma la cita.
* **CU4 – Confirmar asistencia:** El recepcionista valida la identidad del paciente, marca su asistencia y abre la historia clínica.
* **CU6 – Registrar diagnóstico:** El médico registra los datos de la consulta, diagnóstico y observaciones.
* **CU7 – Emitir receta electrónica:** El médico genera una receta asociada al diagnóstico del paciente.
* **CU8 – Solicitar estudio:** El médico solicita estudios adicionales y el sistema genera la orden correspondiente.
* **CU9 – Cargar resultados de estudio:** El técnico ingresa los resultados y el sistema los asocia automáticamente al historial clínico.
* **CU10 – Facturar consulta:** El área administrativa emite la factura correspondiente al seguro médico.
* **CU11 – Registrar pago:** El pago se registra y se vincula a la factura emitida.

## **5. Supuestos del sistema**

* Cada médico posee una agenda de turnos.
* Las historias clínicas son digitales y se asocian a cada paciente.
* Las recetas y órdenes de estudio se generan electrónicamente.
* Los resultados de los estudios quedan disponibles para médico y paciente.
* La facturación se realiza por cada consulta, vinculada al seguro médico correspondiente.

## **6. Diagramas UML a desarrollar**

1. **Diagrama de casos de uso:** mostrará los actores y sus interacciones con el sistema.
2. **Diagrama de clases:** representará las clases principales (Paciente, Médico, Turno, Consulta, Estudio, Factura, etc.) con sus atributos, métodos y relaciones.

## **7. Conclusión**

El análisis realizado permitió identificar los actores, procesos y relaciones más relevantes para el sistema de gestión de una clínica médica.  
 Los diagramas UML resultantes facilitarán el diseño e implementación del software, asegurando una estructura clara y coherente entre los diferentes módulos funcionales.



