

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

PROGRAMACIÓN 3
2da. práctica (tipo b)
(Primer Semestre 2024)

Indicaciones Generales:

- Tiempo estimado: 1h 50 minutos
- Se les recuerda que, de acuerdo al reglamento disciplinario de nuestra institución, constituye una falta grave copiar del trabajo realizado por otro estudiante o cometer plagio para el desarrollo de esta práctica.
- Está permitido el uso de apuntes de clase, diapositivas, ejercicios de clase y código fuente. (Debe descargarlos antes de iniciar con la solución del enunciado)
- No está permitido el uso de entornos de desarrollo integrado (IDEs). Debe utilizar un editor de texto: Notepad++, Sublime, etc. (No es posible utilizar VISUAL STUDIO CODE).
- Está permitido el uso de Internet (únicamente para consultar páginas oficiales de Microsoft y Oracle). No obstante, está prohibida toda forma de comunicación con otros estudiantes o terceros.

PARTE PRÁCTICA (20 puntos)

PUEDE UTILIZAR MATERIAL DE CONSULTA.

Se considerará en la calificación el uso de buenas prácticas de programación (aquellas vistas en clase).

PREGUNTA 1 (20 puntos)

Una reconocida institución clínica de salud en el Perú desea implementar un programa en JAVA que le permita registrar y gestionar toda la información relacionada a las **atenciones médicas** que ofrece. Para esto, lo ha contratado a Ud. con el objetivo de diseñar y realizar la implementación de este programa informático.

Actualmente, la clínica ofrece únicamente **dos tipos** de atenciones médicas: **las citas médicas**, que son encuentros programados entre un profesional médico y un paciente para el diagnóstico y tratamiento de enfermedades; y **los exámenes de diagnóstico**, que son procedimientos programados y realizados por un médico hacia un paciente para identificar una enfermedad, condición o lesión. En ambos tipos de atenciones médicas, se requiere registrar la **fecha y hora de creación de la atención**, la **fecha y hora en que se llevará a cabo la atención médica** y el **estado de la atención** (cuyos valores pueden ser exclusivamente: "PROGRAMADA", "CONFIRMADA", "CANCELADA", "EN_CURSO" y "COMPLETADA"). Asimismo, toda atención médica debe contar con un **identificador único numérico correlativo** que debe ser generado de forma automática por el programa informático al momento de su creación. Así también, se espera que, **durante la creación de una atención médica, se asigne automáticamente la fecha y hora de creación de la misma y se asigne el estado de "PROGRAMADA"**.

Tanto las **citas médicas** como los **exámenes de diagnóstico** tienen datos específicos que se desean manejar dentro del programa informático. Por ejemplo, para las **citas médicas** se desea registrar **el motivo** por el cual el paciente está realizando la programación de una de ellas. Para los **exámenes de diagnóstico** se requiere registrar **el tipo de examen** (que puede ser de tipo: "ECOGRAFIA", "RESONANCIA", "TOMOGRAFIA" y "RAYOS_X").

Con respecto a las **citas médicas**, pueden ser de dos tipos: **consultas presenciales** o **teleconsultas**. En el caso de las **consultas presenciales** se desea registrar si el paciente va a requerir o no **asistencia especial** para su atención presencial. Por otro lado, en el caso de las **teleconsultas** se desea registrar la **plataforma** por donde se llevará a cabo la teleconsulta (que puede ser vía "ZOOM" o "WEBEX", que son las plataformas cuya licencia ha sido adquirida por la clínica), así como el **enlace** de conexión para que el paciente pueda ingresar a la teleconsulta.

Sobre la **consulta presencial** es importante detallar que se lleva a cabo en un **consultorio** de la clínica, mientras que los **exámenes de diagnóstico** se llevan a cabo en una **sala especializada** de la clínica. Tanto el consultorio como la sala especializada son considerados **"ambientes clínicos"** y como tales, se desea manejar algunos **datos tales** como su espacio en **metros cuadrados**, la **torre en la que está ubicado el ambiente clínico** (que es una **letra** puesto que los nombres de las torres de la clínica van desde la A hasta la Z) y el **número de piso**. Sin embargo, existen ciertos datos específicos tanto de los consultorios como de las salas especializadas que también se deben manejar. Los **consultorios** tienen un número de consultorio, mientras que para las **salas especializadas** se desea registrar su nombre y si poseen o no equipamiento de imagenología.

Por otro lado, es importante mencionar tanto **médicos** como **pacientes** poseen DNI, nombre, apellido paterno y apellido materno. No obstante, de los **pacientes** se desea también manejar su **número de historia clínica**. De la misma forma, de los **médicos** se desea manejar su **número de colegiatura**, su **especialidad** y su **número de registro de especialidad**. Con respecto a las especialidades médicas, **únicamente se desea registrar el nombre de la misma**.

Cualquier persona debe devolver información. Por ejemplo, cuando se solicite devolver información a un **paciente**, se devolverá la palabra "**PACIENTE:**" seguido del **DNI**, **nombre**, **apellido paterno** y **apellido materno**. Asimismo, cuando se solicite devolver información a un **médico**, se devolverá la palabra "**MEDICO:**" seguido del **nombre**, **apellido paterno**, **apellido materno**, la palabra "**CMP:**", **el número de colegiatura**, la palabra "**ESPECIALIDAD:**" y el **nombre de la especialidad**. De igual manera, cualquier ambiente clínico deberá devolver información. En el caso de una **sala especializada**, se deberá devolver la palabra "**SALA ESPECIALIZADA:**", el **nombre de la sala**, la palabra "**TORRE:**", seguido de **la letra que representa la torre** donde está ubicada, la palabra "**PISO:**", seguido del **número de piso** donde está ubicada la sala, la palabra "**EQ. IMAGENOLOGIA:**" y la indicación de si cuenta o no con equipamiento de imagenología. Para el caso de un consultorio, se deberá devolver la palabra "**CONSULTORIO:**", seguido del **número del mismo**, la palabra "**TORRE:**", seguido de **la letra que representa la torre** donde está ubicado, la palabra "**PISO:**", seguido del **número de piso** donde está ubicado el consultorio.

Finalmente, las **atenciones médicas** deben ser consultables. Para todo tipo de atención, se devolverá la palabra "**ID:**", seguido del **identificador**, la palabra "**CREACION:**", seguido de **la fecha y hora de creación** de la atención, la palabra "**ATENCION:**", seguido de **la fecha y hora de la atención**, seguido del **estado** de la misma, los **datos del médico** y los **datos del paciente**. En el caso de **atenciones presenciales** se agregará los **datos del consultorio**, la palabra "**EQ. IMAGENOLOGIA:**" y un **indicador de si se requiere o no asistencia** para el paciente. En el caso de **exámenes de diagnóstico**, se agregará los datos de la **sala especializada**, la palabra "**TIPO EXAMEN:**" y el **tipo de examen**. En el caso de **teleconsultas**, se agregará la palabra "**PLATAFORMA:**", el nombre de la plataforma por donde se llevará a cabo la misma, la palabra "**ENLACE:**" y la URL del enlace de conexión.

Se ha realizado un análisis preliminar y se han detectado algunas clases de manera genérica:

- **Especialidad:** Clase que define los atributos y comportamientos de la especialidad de un médico.
- **Persona:** Clase que define los atributos y comportamientos comunes de los médicos y pacientes.
- **Paciente:** Clase que define los atributos y comportamientos de los pacientes. Asimismo, define el método *listarCitasMedicasProgramadas()* de tipo void que permite imprimir únicamente todas las atenciones médicas de tipo cita médica, es decir consultas presenciales y teleconsultas, más no exámenes de diagnóstico.
- **AtencionMedica:** Clase que define los atributos y comportamientos comunes de los exámenes de diagnóstico y las citas médicas. Asimismo, define el método "consultarDatos" de tipo String y abstracto.
- **CitaMedica:** Clase que define los atributos y comportamientos comunes de las consultas presenciales y las teleconsultas.
- **AmbienteClinico:** Clase que define los atributos y comportamientos comunes de los consultorios y las salas especializadas.
- **IConsultable:** Clase de tipo interface que define el método "devolverInformacion" de tipo String y la obligación a las personas y ambientes clínicos de devolver información detallada sobre los mismos.

Se le solicita elaborar un programa en **JAVA** que permita dar soporte a la lógica de negocio mencionada y además permita imprimir el reporte que se muestra en la Figura 02. El reporte debe verse igual.

Para validar el modelado de clases a realizar y como parte de las pruebas del sistema, se cuenta con el código fuente de la clase Principal, cuya programación se encuentra definida en la Figura 01. La programación de la clase Principal debe ser exactamente igual al que se está colocando en la Figura 01.

```

import java.text.SimpleDateFormat;
import java.util.ArrayList;
class Principal{
    /* Coloque sus datos
    Nombre Completo:
    Código PUCP:
    */
    public static void main(String[] args) throws Exception{
        //Se crean tres especialidades
        Especialidad esp01 = new Especialidad("MEDICINA INTERNA");
        Especialidad esp02 = new Especialidad("RADIOLOGIA");
        Especialidad esp03 = new Especialidad("OTORRINOLARINGOLOGIA");
        //Se crean tres médicos de distintas especialidades
        Medico med01 = new Medico("18282901", "NATALI", "LEIVA", "REYES", esp01,
"062258", "030568");
        Medico med02 = new Medico("13876114", "RAUL", "NEIRA", "GORRITI", esp02,
"059092", "030077");
        Medico med03 = new Medico("29813127", "LUIS", "PEREZ", "LU", esp03,
"039811", "029899");

        //Se crea una paciente
        Paciente pac01 = new Paciente("17920084", "PATRICIA", "FERNANDEZ", "TELLO",
3873);
        //Se crea un consultorio
        Consultorio con01 = new Consultorio(17.8, 'A', 2, "204");
        //Se crea una sala especializada
        SalaEspecializada sal01 = new SalaEspecializada(39.8, 'B', 1, "CAYETANO",
true);
        //Creamos una atencion medica de tipo Consulta Presencial para el 29 de Marzo
a las 16:00
        SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy HH:mm");
        AtencionMedica atm01 = new ConsultaPresencial(pac01, med01, sdf.parse("29-03-
2024 16:00"), "DOLOR DE ESPALDA", con01, false);
        //Creamos una atencion medica de tipo Examen Diagnostico para el 30 de Marzo
a las 08:30
        AtencionMedica atm02 = new ExamenDiagnostico(pac01, med02, sdf.parse("30-03-
2024 08:30"), sal01, TipoExamen.ECOGRAFIA);
        //Creamos una atencion medica de tipo Teleconsulta para el 01 de Abril a las
15:30
        AtencionMedica atm03 = new Teleconsulta(pac01, med03, sdf.parse("01-04-2024
15:30"), "DOLOR DE GARGANTA", Plataforma.ZOOM, "https://pucp.zoom.us/j/98403730464");
        //Relacionamos las atenciones con el paciente
        pac01.setAtencionesMedicas(new ArrayList<>());
        pac01.getAtencionesMedicas().add(atm01);
        pac01.getAtencionesMedicas().add(atm02);
        pac01.getAtencionesMedicas().add(atm03);
        //Consultamos datos de todas las atenciones médicas
        System.out.println("PRIMERA CONSULTA....");
        System.out.println(atm01.consultarDatos());
        System.out.println(atm02.consultarDatos());
        System.out.println(atm03.consultarDatos());
        //Consultamos datos de únicamente las citas médicas del paciente 01
        System.out.println("SEGUNDA CONSULTA....");
        pac01.listarCitasMedicasProgramadas();
    }
}

```

el arraylist debio estar dentro del constructor de la clase -> ya que rompe el encapsamiento

Fig. 01. Código Fuente de la clase Principal

```

C:\Users\PC_Lima\OneDrive\LP2\2024-1\Laboratorio 02\Solucion\JAVA>javac *.java

C:\Users\PC_Lima\OneDrive\LP2\2024-1\Laboratorio 02\Solucion\JAVA>java Principal
PRIMERA CONSULTA.....
ID:1 - CREACION:10-03-2024 14:00 - ATENCION:29-03-2024 16:00 - PROGRAMADA
MEDICO: NATALI LEIVA REYES - CMP:062258 - ESPECIALIDAD:MEDICINA INTERNA
PACIENTE: 17920084 - PATRICIA FERNANDEZ TELLO
CONSULTORIO:204 - TORRE:A - PISO:2
REQ. ASISTENCIA:NO

ID:2 - CREACION:10-03-2024 14:00- ATENCION:30-03-2024 08:30 - PROGRAMADA
MEDICO: RAUL NEIRA GORRITI - CMP:059092 - ESPECIALIDAD:RADIOLOGIA
PACIENTE: 17920084 - PATRICIA FERNANDEZ TELLO
SALA ESPECIALIZADA:CAYETANO - TORRE:B - PISO:1 - EQ. IMAGENOLOGIA:SI
TIPO EXAMEN:ECOGRAFIA

ID:3 - CREACION:10-03-2024 14:00 - ATENCION:01-04-2024 15:30 - PROGRAMADA
MEDICO: LUIS PEREZ LU - CMP:039811 - ESPECIALIDAD:OTORRINOLARINGOLOGIA
PACIENTE: 17920084 - PATRICIA FERNANDEZ TELLO
PLATAFORMA:ZOOM - ENLACE:https://pucp.zoom.us/j/98403730464

SEGUNDA CONSULTA.....
ID:1 - CREACION:10-03-2024 14:00 - ATENCION:29-03-2024 16:00 - PROGRAMADA
MEDICO: NATALI LEIVA REYES - CMP:062258 - ESPECIALIDAD:MEDICINA INTERNA
PACIENTE: 17920084 - PATRICIA FERNANDEZ TELLO
CONSULTORIO:204 - TORRE:A - PISO:2
REQ. ASISTENCIA:NO

ID:3 - CREACION:10-03-2024 14:00 - ATENCION:01-04-2024 15:30 - PROGRAMADA
MEDICO: LUIS PEREZ LU - CMP:039811 - ESPECIALIDAD:OTORRINOLARINGOLOGIA
PACIENTE: 17920084 - PATRICIA FERNANDEZ TELLO
PLATAFORMA:ZOOM - ENLACE:https://pucp.zoom.us/j/98403730464

```

Fig. 02. Salida del sistema informático

Es indispensable que se encuentren programados todos los atributos de las clases y sus relaciones que permiten la impresión del reporte. **Con respecto a los constructores, getters y setters, puede programar solo aquellos que son requeridos para la salida del reporte.**

Se ha compartido la clase Principal.java en PAIDEIA. Una vez finalizada la evaluación, suba todo su código fuente a PAIDEIA en un archivo .zip que tenga como nombre su código PUCP.

Aspectos a considerar para evitar descuento de puntos:

- Colocar su nombre completo y código PUCP únicamente en la clase Principal.
- Nombrar correctamente a las clases, atributos y métodos.
- No realice importaciones innecesarias de clases (no utilice * para las importaciones).
- Utilice los principios de POO: abstracción (en clases y métodos donde sea requerido), polimorfismo y encapsulamiento.
- Utilice correctamente ámbitos, clases / métodos abstractos(as), clases de tipo interface, enumerados donde sea requerido.
- Utilice la etiqueta "@Override" donde corresponda.
- Respetar el orden en la estructura de una clase.
- Emplear un archivo por clase.
- El programa debe compilar correctamente, se descontarán puntos por errores de compilación.
- Utilice los métodos solicitados para la impresión del reporte (optimice el código).

Rúbrica de calificación:

(0.5 puntos) Correcta programación de la clase de tipo interface.

(0.5 puntos) Correcta programación de las clases de tipo enumerate.

(9.0 puntos) Correcta programación de todas las clases con sus atributos, relaciones, constructos, *getters* y *setters* (respetando encapsulamiento, abstracción (métodos y clases), herencia, interfaces y enumerados). **[A pesar de que las impresiones no se realicen, si se cuenta con todas las clases y estas poseen todos los atributos y relaciones programadas, recibe los 9.0 puntos de este apartado].**

(4.0 puntos) Correcta programación del método *devolverInformacion()* en las clases donde es requerido el uso de la interface.

(4.0 puntos) Correcta programación del método *consultarDatos()* en las clases donde es requerido.

(2.0 puntos) Correcta programación del método *listarCitasMedicasProgramadas()* en la clase Paciente.