

# Desarrollo de aplicación utilizando conceptos POO con Java

Actividad de programación orientada a objetos, una aplicación desarrollada en Java con el paradigma de programación orientado a objetos.

Autor:

Ethan Sarricolea

## Índice

<b>Desarrollo de aplicación utilizando conceptos POO con Java.....</b>	<b>1</b>
<b>Índice.....</b>	<b>1</b>
<b>Instrucciones.....</b>	<b>1</b>
<b>Entrega.....</b>	<b>2</b>
<b>Descripción.....</b>	<b>2</b>
Entidades.....	2
Hospital.....	2
Person.....	2
Doctor.....	2
Paciente.....	3
Cita.....	3
Expediente médico.....	3
Relaciones.....	3
Sistema.....	3
<b>Diagrama UML.....</b>	<b>4</b>

## Instrucciones

Diseñar una aplicación aplicando la POO con Java de uno de los temas que se mencionan a continuación:

1. Sistema de reserva de aerolíneas/autobuses/trenes
2. Sistema de gestión de bibliotecas
3. Sistema de atención médica
4. Sistema de seguimiento de vehículos
5. Sistema de banca electrónica
6. Sistema de nómina
7. Otra (solicitar autorización del profesor)

# Entrega

- Describe la aplicación a desarrollar, incluir menú por consola. (PDF)
- Desarrollar el diagrama UML (paquetes, herencia, otras relaciones (mismo PDF anterior)
- Construye la aplicación en lenguaje de programación Java (código fuente .zip)
- Utiliza control de versiones, sube la aplicación en GitHub (enlace Github))

## Descripción

Enunciado: Se quiere realizar un sistema de atención médica, donde se gestionan los hospitales, sus médicos, pacientes y citas.

## Entidades

### Hospital

El hospital es la institución donde se realizarán las atenciones médicas, del hospital es necesario saber cuantos hospitales hay, su nombre y su identificador único alfanumérico. De acorde al nivel del hospital tendrán diversas áreas y serán las atenciones que podrá dar.

El hospital tiene una lista de pacientes, citas y doctores.

En el hospital se agendarán las citas, se agregaran los pacientes y además se podrán ver los listados.

### Person

Esta clase persona, será una clase abstracta acerca de las personas. en ella se registraran datos como el nombre, fecha de nacimiento y sexo, ademas de tener los metodos concretos para la obtencion de estos mismos datos y su modificacion (Setters y getters).

### Doctor

Los médicos son una extensión de la clase abstracta persona, ya que tienen todos sus datos de identificación personal y métodos de edición de estos. Además de ello los médicos / doctores tienen un atributo especialidad, que determina en qué zona se especializan (ya sea laboratorio, enfermería, cirugía, etc.).

## Paciente

Los pacientes igualmente se extienden de la clase persona, por lo que se identifican por nombre, fecha de nacimiento y sexo, además tienen un atributo "hasExpediente" que determina si tienen o no un expediente.

## Cita

Los médicos atenderán a los pacientes en las citas, cada cita se registra en el sistema y contiene un identificador único, la fecha de agenda, el hospital, el área y nombre de paciente.

## Relaciones

- Los médicos y pacientes son personas
- El hospital necesita médicos que trabajen en el
- El hospital puede agendar citas
- Una cita requiere de un paciente

## Sistema

El sistema permite ver la información del hospital, el listado de empleados y el listado de clientes.

1. El sistema debe mostrar el menú de opciones para ver la información del hospital, ver información sobre los empleados, de los pacientes, de las citas o cerrar el sistema.
2. El sistema debe permitir ver y registrar doctores / personal que trabaje en el hospital
3. El sistema debe permitir ver la información del hospital, incluido el conteo que lleva de pacientes
4. El sistema debe permitir ver la lista de pacientes registrados en alguna cita
5. El sistema debe permitir agendar una cita con un paciente, doctor y fecha y/o ver las citas agendadas

## Diagrama UML

