**Sistema de Gestor de Pacientes.  
  
public class Usuarios  
{  
String asistente;  
String Medicos;  
String administradores;  
  
}**

**public class Usuarios  
{  
String rol; --El valor de rol solamente puede ser administrador, medico o asistente.   
}**

**interface Mantenimientos  
  
{  
public agregar();  
public modificar();  
public Eliminar();   
  
}  
  
class MantenimientoUsuarios (La clase Mantenimiento Usuario hereda de la clase  
usuario e implementa la interfaz mantenimiento).{**

**Mantenimiento usuario no puede heredar de usuario pork romperia el principio SRP. Pero si se relaciona con usuario por una composicion.**

**El mantenimiento usuario tiene 3 roles que son asistentes, administrador y medico.  
empezando por:  
  
class ControladorAsistente ~~extends M~~antenimiento~~Medicos~~ implement Mantenimiento (La clase que controla el acceso a datos para hacer el mantenimiento(agregar, modificar, eliminar) no hereda, simplemente implementa la interface.  
{  
agregar~~.Paciente~~(); --Dentro de esta clase van los metodos que se implementan de la inferface. Agregar no se define en la clase Paciente sino en esta clase.  
~~agregar.Medicos()~~; --Lo mismo**

**\_\_\_**ListadoPaciente;  
BusquedaPaciente;  
LitadoMedico.especialidad;  
~~Mantenimiento de citas~~ **-- El mantenimiento de Citas no se hace en esta clase.   
\_\_\_**

**Nota:  
El asistente solo puede agregar pacientes y medicos por especialidad por lo que estara a parte  
de implementar la interfaz de mantenimiento, tambien extendera del mantenimiento de medicos  
  
}  
  
class Medicos  
{  
"Segun el documento del proyecto dice que los medicos tienen acceso a todo con excepcion  
del mantenimiento usuario, por lo que le pondremos las mismas acciones que el asistente"  
  
"Estos son los mantenimientos a los que los medicos tienen acceso, seran tratados  
como objetos en esta clase para que cuando se extiendan a la otra clase se puedan  
desarrollar"   
  
MantenimientoPacientes;  
MantenimientoCitas;  
MantenimientoPadecimientos;   
MantenimientoRecetas;   
MantenimientoPruebasLaboratorio;   
MantenimientosResultadosPruebasLab;   
LitadoMedico.especialidad;  
Búsqueda de Pacientes;  
Historial del paciente;  
Impresión de Recetas;  
Estadísticas;  
  
}  
  
class Administradores  
{  
"Los administradores tienen acceso a toda la aplicacion"  
  
"Estos son los mantenimientos a los que los medicos tienen acceso, seran tratados  
como objetos en esta clase para que cuando se extiendan a la otra clase se puedan  
desarrollar"   
  
MantenimientoUsuarios;  
MantenimientoMedicos;  
MantenimientoEspecialidad   
MantenimientoAsistentes   
MantenimientoPacientes;  
MantenimientoCitas;  
MantenimientoPadecimientos;   
MantenimientoRecetas;   
MantenimientoPruebasLaboratorio;   
MantenimientosResultadosPruebasLab;   
LitadoMedico.especialidad;  
Búsqueda de Pacientes;  
Historial del paciente;  
Impresión de Recetas;  
Estadísticas;  
ListadoPaciente;  
BusquedaPaciente;  
LitadoMedico.especialidad;  
Mantenimiento de citas  
  
}  
  
class ~~Mantenimiento~~Medicos ~~implement Mantenimiento~~ --La clase medico no puede implementar el mantenimiento porque es el Modelo. Quien debe implementar Mantenimiento es la clase que maneja el acceso a datos ControladorMedico.  
{  
  
int CodEmpleado;  
String nombre;  
String apellido;  
String telefono;  
String direccion;  
String cedula;  
String especialidad;  
  
  
}  
  
class MantenimientoEspecialidad ~~extends MantenimientoMedicos~~ implement Mantenimiento  
{  
  
int codigo;  
String nombre;  
}  
  
class MantenimientoAsistentes extends Usuarios implement Mantenimiento  
{  
  
int codEmpleado;  
String nombre;  
String apellido;  
String telefonos;  
String direccion;  
String cedula;  
  
}  
  
class MantenimientoPacientes implement Mantenimientos  
{  
String nombre;  
String apellido;  
String telefono;  
String direccion;  
String cedula;  
String fechaNacimiento;  
String fumador;  
String alergias;  
String foto;**

**Mas o menos entendi como esta puesto eso. Algunas partes del modelo estan juntas con algunas del controlador. Pero deben ir separadas.**

**Public void BusquedaPaciente(String nombre, String apellido, String cedula) –Dentro del controlador de pacientes.  
  
nota:  
una foto de cada paciente se desplegará al seleccionar un paciente de la tabla)  
  
HistorialPaciente (se visualizará informaciones del paciente, tales como:  
fechas de visitas, causas de las visitas, observaciones, recetas y resultados de  
pruebas de laboratorio ) esto es un iReport.  
}  
  
class MantenimientoCitas extends MantenimientoPacientes   
{  
  
String paciente;  
String fecha;  
String hora;  
String medico;  
String causa;  
  
}  
  
class MantenimientoPadecimientos extends MantenimientoPacientes   
{  
  
int codigo;  
String nombre;  
Estadísticas (Top 10 padecimientos más frecuentes Top 10 médicos más  
consultados).   
}  
  
class MantenimientoRecetas extends MantenimientoPacientes   
{  
  
String paciente;  
String padecimiento;  
String medicamentos;   
  
Impresión de Recetas esto es un iReport.  
  
}  
  
class MantenimientoPruebasLaboratorio   
{  
int codigo;  
String nombrePrueba;  
  
}  
  
class MantenimientosResultadosPruebasLab extends MantenimientoPacientes   
{  
  
String paciente;  
String prueba;  
String resultado;  
  
}**