

**ITLA – Tecnólogo en Desarrollo de Software
Programación II.**

Proyecto Final (Sistema Gestor de Pacientes)

Profesor: Raydelto Hernández Perera.

Sistema Gestor de Pacientes

Objetivos específicos:

- Fomentar el trabajo en equipo.
- Evaluar el dominio del paradigma de orientación a Objetos
- Calificar la capacidad de segmentar en capas la complejidad de un sistema.
- Evaluar el uso adecuado de patrones de diseño que faciliten el mantenimiento de la estructura del sistema
- Evaluar capacidades de análisis y diseño (deben de documentar todo antes de empezar a codificar).

Descripción del proyecto

Cree un sistema para gestionar las visitas de pacientes de un hospital. A dicho sistema podrán acceder los médicos y las asistentes.

El sistema permitirá a los médicos ingresar pacientes al sistema así como mantener un historial de citas y recetas, así como los reportes descritos en la sección de funcionalidades.

Permitirá a las asistentes (ver y buscar [sin acceso a modificar]) el listado de pacientes, listado de visitas, y listado de recetas

Los usuarios de tipo administrador tendrán acceso al sistema completo, y es el único usuario que tiene permiso para crear usuarios.

Funcionalidades del usuario administrador

- Mantenimiento de usuarios (3 roles de usuarios son requeridos: Administrador, Asistente y Médico) [Las asistentes solo tienen acceso a agregar Paciente, y listado de médicos], [Los médicos tienen acceso a todo, menos al mantenimiento de usuarios, y los administradores tienen acceso a todo].

- Mantenimiento de Médicos (Código de empleado, Nombre, apellido, teléfonos, dirección, cédula, Especialidad).
- Mantenimiento de Especialidades (Código y Nombre).
- Mantenimiento de Asistentes (Código de empleado, Nombre, apellido, teléfonos, dirección, cédula)

+ (todas las funcionalidades de asistentes y médicos)

Funcionalidades del usuario con acceso de médico

- Mantenimiento de Pacientes (Nombre, apellido, teléfono, dirección, cédula, fecha de nacimiento, fumador, alergias, foto).
- Mantenimiento de Citas (Paciente, fecha, hora, médico, causa).
- Mantenimiento de Padecimientos (Código y Nombre).
- Mantenimiento de Recetas (Paciente, padecimiento, medicamentos).
- Mantenimiento de Pruebas de Laboratorio (Código, nombre de la prueba)
- Mantenimiento de Resultados de Pruebas de Laboratorio (Paciente, prueba y resultado)
- Listado de Médicos por especialidad
- Búsqueda de Pacientes, (búsqueda por nombre, apellido y/o cédula, una foto de cada paciente se desplegará al seleccionar un paciente de la tabla)
- Historial del paciente (se visualizará informaciones del paciente, tales como: fechas de visitas, causas de las visitas, observaciones, recetas y resultados de pruebas de laboratorio)
- Impresión de Recetas (Impresión en soporte de papel de la receta).
- Estadísticas (Top 10 padecimientos más frecuentes, Top 10 médicos más consultados).

Funcionalidades del usuario con acceso de asistente

- Listado de pacientes
- Búsqueda de Pacientes, (búsqueda por nombre, apellido y/o cédula, una foto de cada paciente se desplegará al seleccionar un paciente de la tabla)
- Listado de Médicos por especialidad
- Mantenimiento de citas

Aclaraciones:

- Entiéndase **mantenimiento** como parte del sistema que permite (agregar, modificar y eliminar una entidad).
- No asuma **NADA** que no esté descrito en este documento sin antes consultar con su instructor.

Evaluación:

El objetivo de esta asignación es demostrar que se dominan los conocimientos de la asignatura Programación II.

Detalle de aspectos a evaluar:

Análisis y Diseño orientado a Objetos (5 puntos)

En vista de que este es un curso de diseño orientado a objetos debe de presentarse el diagrama UML de clases así como el diagrama de base de datos (database diagram) fruto del proceso de diseño del sistema.

Enseñar los diseños, diagramas y prototipos al instructor antes de empezar a codificar es altamente recomendado.

Funcionalidades del sistema. (5 puntos)

En este punto se evaluará que tanto se corresponden las funcionalidades requeridas con las funcionalidades implementadas.

Si el sistema no sirve para nada útil pues el grupo obtendrá un **CERO** en el proyecto entero. *Ejemplo: (De nada me sirve que el login funcione correctamente si no puedo hacer nada útil con el sistema).*

Patrones de diseño (5 puntos)

Debe de implementarse satisfactoriamente al menos tres patrones de diseño para enfrentar problemas comunes de diseño

Segmentación de la complejidad (2.5 puntos).

La complejidad del sistema debe de dividirse en al menos 2 capas (Modelo y Vista), de modo que sea más manejable el sistema para fines de darle mantenimiento así como para agregar nuevas funcionalidades.

*Ejemplo: Si su sistema presenta una consulta a una base de datos desde una clase que extiende de JFrame usted tendrá **cero** en este renglón, y tal error repercutirá en su calificación del renglón de diseño.*

Usabilidad (2.5 puntos)

El sistema debe de ser intuitivo y de fácil uso. Cualquier persona pudiera saber como usarlo solamente de ver las opciones de menú.

NOTA: En la asignatura de diseño centrado en el usuario profundizarán más sobre este tópico, no obstante, la creación de aplicaciones de uso intuitivo es un objetivo de este curso.

Referencias: <http://es.wikipedia.org/wiki/Usabilidad>