

Proyecto #1

Prof. Karol Leitón Arrieta
Prof. Jennifer Fuentes Bustos
Prof. Maikol Guzmán Alán
Prof. Marianela Solano Oreas
Prof. Darío Ríos Navarro

Fecha de entrega: Jueves 12 de setiembre de 2019
Modo de entrega: Aula Virtual Institucional
Modo de trabajo: Individual

Objetivos

- Aplicar los conceptos teóricos, principios y técnicas estudiadas en clase.
- Investigar e incorporar conceptos y técnicas.
- Desarrollar código correctamente estructurado, debidamente encapsulado, eficiente y fácilmente reutilizable

Generalidades

- El programa consiste en un sistema de carácter administrativo para un hospital, se trata de un programa estándar de consola, que permita al usuario ejecutar cada una de las funciones necesarias. Se debe hacer uso y manejo adecuado de colecciones y relaciones.
- El código del proyecto debe estar estructurado adecuadamente, respetando los principios de diseño estudiados en clase.
- Las clases deben separar correctamente la declaración de la interfaz y su implementación, usando archivos de cabecera (archivos .h) y código fuente (archivos.cpp) por aparte. Los archivos de cabecera deberán siempre usar guardas (no se permite el uso de la directiva #pragma once cuando se utilice Visual Studio).
- Las clases de entidad no deberán contener código de entrada/salida (como salida a la consola usando cout, por ejemplo)
- El manejo de la interfaz con el usuario debe hacerse por medio de una o varias clases diseñadas para tal efecto.
- La interfaz debe ser diseñada en forma agradable, entendible y fácil de utilizar por cualquier usuario., recuerde que si el usuario no entiende el sistema posiblemente se trate de un problema de diseño por parte del programador. Pruebe su programa con otros usuarios, esto para recibir retroalimentación y asegurarse que la interfaz sea clara y entendible.
- El programa debe tener una interfaz clara y un adecuado e intuitivo manejo de menús y submenús, que faciliten el uso de la interfaz, de lo contrario el director del hospital no aprobará la compra del sistema.
- El sistema debe tener un adecuado y excelente manejo y control de errores, el cual no permita incluir datos diferentes al que se está solicitando, si el programa se “cae” el director del Hospital rechazará la compra del sistema en forma definitiva y buscará otro desarrollador de software.
- Debe presentar una funcionalidad correcta de los módulos mediante una codificación eficiente y bien estructurada.

- Debe presentarse un establecimiento correcto de relaciones (por ejemplo: no se puede ingresar una cama si no existe un pabellón, no se puede ingresar un paciente si no existe una cama o no se puede crear un doctor si no existe la especialidad)
- Incluir UML de clases, el cual concuerde con el código (no tiene puntaje pero debe entregarse)

Descripción

Se requiere realizar un sistema que permita administrar el manejo de personal y pacientes en el área de cirugías del Hospital Esperanza.

(Sobre Especialidades)

El hospital contempla un máximo de 10 **especialidades** (neumología, hematología, ortopedia, etc.), por lo cual el sistema debe permitir la inclusión de los datos de cada especialidad.

(Sobre Doctores)

El hospital debe llevar registro de todos sus doctores, por política pública el número de doctores contratados no podrá ser mayor a 20. Para registrar un doctor es indispensable que exista dicha especialidad en la lista de especialidades del hospital, de lo contrario no se podrá registrar el doctor.

(Sobre Pabellones)

El hospital cuenta con un máximo de 10 pabellones cada uno identificado por una letra. Cada pabellón es destinado para un género en particular, es decir hay pabellones solo para hombres y otros solo para mujeres. En cada pabellón cabe un máximo de 20 camas.

(Sobre Camas)

El sistema debe permitir la inclusión de **camas**, cada cama que se ingrese debe ser asignada a un pabellón específico.

Cada cama tiene un código único, definido por la unión entre la letra del pabellón y el número de cama, por ejemplo: cama A8.

Cada cama tiene además un estado: ocupado o disponible.

(Sobre Pacientes)

En cuanto a la cantidad de pacientes, el hospital tiene una cantidad máxima de 200.

Cada vez que se interne un paciente, es importante tomar sus datos personales tales como:

- Cédula, nombre, apellidos, dirección, patología, tipo de cirugía
- estado: internado o dado de alta
- Prioridad: baja, media, crítica
- fecha de su cirugía
- Cama (debe existir en el sistema)
- Médico (debe existir en el sistema)

Si no hay camas disponibles el paciente no puede ser ingresado.

***Nota:** recordar no ingresar ningún registro repetido, por ejemplo camas repetidas, doctores o pacientes repetidos.*

Tabla de Evaluación

Nota: **En cada ítem se evalúa funcionalidad y eficiencia del código**

Ítem	Puntos	Puntos Obt	Comentarios
Inclusión de pabellones <i>(no requiere ninguna relación particular)</i>	1 pts	0	
Inclusión, visualización y eliminación adecuada de camas . <i>Recuerde: no se puede ingresar una cama si no existe el registro previo de pabellones. No se puede eliminar una cama si tiene asignado un paciente. <u>Respetar relaciones requeridas para la clase</u></i> <i>Se debe poder visualizar de manera general la lista de todas las camas organizadas por pabellón, incluyendo código de cama, estado y nombre del paciente (en caso de estar ocupada)</i>	15 pts	0	
Inclusión, visualización y eliminación adecuada de doctores . <i>Recuerde: no se puede ingresar un doctor si la especialidad no ha sido previamente ingresada, no se puede eliminar un doctor si tiene pacientes asignados. <u>Respetar relaciones requeridas para la clase</u></i> <i>Se debe poder visualizar en forma general la lista de todos los doctores incluyendo: nombre, especialidad y <u>listado de todos sus pacientes</u>.</i>	22 pts	0	
Inclusión, visualización y eliminación adecuada de Pacientes . <i>Recuerde: no se puede ingresar pacientes si no hay camas disponibles.</i> <i>A cada paciente se le debe asignar una cama y un doctor, así como todos los demás datos requeridos para su ingreso.</i> <i>Se debe visualizar en pantalla la lista de los pacientes incluyendo su id, nombre, cama y medico</i>	17pts	0	
Cambio de cama para un paciente <i>El sistema debe dar la opción de cambiar la cama asignada a un paciente específico. También se debe dar la opción de liberar la cama asignada a un paciente en caso de que a este se le haya dado de alta.</i>	6pts		

Cambio de médico para un paciente <i>El sistema debe dar la opción de cambiar el medico asignado a un paciente específico.</i>	6pts		
Informe por paciente <i>Se debe permitir buscar un paciente específico y mostrar un informe con los siguientes datos: nombre, id, doctor, pabellon, estado, cama (en caso de estar internado), especialidad y patología</i>	15pts		
Validación de errores, manejo correcto de las salidas y entradas, usabilidad, organización offline, manejo correcto de memoria	18 pts	0	
Total	100	0	

Observaciones generales:

- Cada profesor define el IDE y la versión del mismo en que se deberá entregar el proyecto.
- El medio oficial de entrega será el aula virtual institucional.
- Se revisa el archivo subido al aula virtual, por lo cual verifique bien haber subido la versión correcta pues no se recibirá ningún otro archivo o versión después de la fecha de entrega.
- Queda a criterio del profesor solicitar defensa del proyecto.
- No se admitirá la entrega tardía del proyecto, en este caso, se calificará con nota 0 (cero).
- Los proyectos deben entregarse con código fuente completo y compilable así como su respectivo UML en pdf.
- Cualquier entrega que no sea de elaboración original del estudiante o haya sido copiado o adaptado de otro origen (plagio), de manera parcial o total, se calificará con nota 0 (cero) y se podrá proceder como lo indiquen los reglamentos vigentes de la universidad para elevar el caso.
- **El código del programa debe compilar correctamente, si el código no puede compilarse, el trabajo será calificado con una nota de 0 (cero).**
- Se evalúa la funcionalidad y eficiencia del programa, si algo NO funciona NO puede ser eficiente, por tanto lo que no funciona no se evalúa.