

Gestión de Procesos Prioritarios en un Policlínico

Docente

Bases de Datos II - Ingeniería de Software
Curso: 2025 - 2026

El presente documento describe una problemática para ser resuelta durante el semestre. Presenta además, particulares a evaluar por cada asignatura de manera independiente y en conjunto.

Características generales

El proyecto tributa a la evaluación de las asignaturas Bases de Dato II e Ingeniería de Software, donde cada una controlará aspectos no necesariamente coincidentes.

El desarrollo de la solución será llevado a cabo por un equipo de estudiante de hasta 5 integrantes, aunque la cantidad exacta de miembros se confirmará durante las primeras clases presenciales.

La nota de cada miembro del equipo será individual y no necesariamente la misma para ambas asignaturas, teniendo en cuenta:

- Labor de curso.
- Trabajo del equipo.
- Requisitos particulares de la asignatura de Bases de Datos (ver archivo adjunto).
- Requisitos particulares de la asignatura de Ingeniería de Software (al final del documento).

Descripción del problema

En un policlínico docente se manejan diariamente múltiples procesos que abarcan desde la atención a pacientes hasta la gestión interna de los recursos. Actualmente, estos procesos se realizan de manera parcial o totalmente manual, lo que genera pérdida de tiempo, duplicación de datos y dificultades en la trazabilidad de la información.

Para solucionar esta problemática, se propone el desarrollo de una aplicación web integral que permita centralizar y automatizar la gestión de los principales procesos del policlínico. Esta herramienta deberá incorporar bases de datos robustas y seguras que gestionen información clave con el objetivo de optimizar el trabajo diario y facilitar la comunicación entre áreas, integrando los flujos de trabajo.

En el centro labora personal de salud conformado por doctores, enfermeros, técnicos, farmacéuticos y otros especialistas. Cada trabajador queda registrado con datos como nombre, identificación, cargo, departamento al que pertenece y estado laboral; considere que cada departamento cuenta con un jefe responsable de su gestión, el cual labora allí. La herramienta debe permitir registrar y actualizar al personal, asignarlo a un departamento específico y designar jefes cuando corresponda.

Los departamentos ofrecen consultas programadas y atención de urgencias en el Cuerpo de Guardia. Las consultas pueden generarse solo por remisión previa desde un puesto médico externo o por derivación desde otro departamento del mismo policlínico.

De cada paciente se conoce su nombre completo, dirección, número de identidad, edad y contacto. El sistema debe almacenar y mantener actualizado el historial clínico de cada paciente, centralizando todos los registros generados en diferentes departamentos. Cuando un paciente asiste a consultas, ya sea programadas o de urgencias, el doctor principal registra los detalles de la atención, incluyendo el médico tratante, diagnóstico, fecha, medicamentos recetados y el departamento donde se realizó la consulta. Además, el sistema debe poder listar consultas por tipo en rangos de fechas específicos, mostrar las últimas diez atenciones con detalles del paciente, el médico y el departamento, y permitir el acceso al historial clínico completo cuando sea necesario para dar continuidad a tratamientos por parte de cualquier personal de la salud.

Los registros por departamento, incluyendo el Cuerpo de Guardia, se gestionan de manera independiente. Sin embargo, al consultar la información de un paciente, debe ser posible seleccionar el o los departamentos deseados y, además, fusionar los registros de varios departamentos. De esta forma, el sistema permitirá visualizar de manera unificada todos los datos del paciente en orden cronológico.

En el Cuerpo de Guardia, por cada atención realizada, el personal que trabaja ese día registra los datos del paciente (nombre, dirección, edad, número de identidad) y el médico prescribe un tratamiento que pasa al departamento de enfermería. La enfermería actualiza el registro basándose en el método prescrito, indicando los medicamentos suministrados antes de que el paciente los reciba y lo descuenta de su *stock* de medicamentos.

Cada departamento maneja su propio *stock* de medicamentos, cuya actualización y modificaciones deben ser aprobadas por la dirección del policlínico. Diariamente se realiza un conteo de medicamentos para actualizar el inventario y solicitar la reposición correspondiente al almacén central, previa autorización de la dirección médica (jefe de departamento) y las jefaturas pertinentes (dirección del policlínico).

El almacén central concentra el mayor *stock* de medicamentos del policlínico. Cada *stock* cuenta con la lista de medicamentos que puede almacenar y, de cada medicamento, se cuenta con su nombre comercial, nombre científico, lote de creación, fecha de caducidad, empresa comercializadora que lo creó y su formato (jarabe, cápsulas, etc.). Ante solicitudes de reposición, el jefe de almacén analiza los niveles máximos y mínimos de cada medicamento y asigna cantidades a los departamentos. Además, mensualmente, el personal de cada departamento registra el consumo acumulado de medicamentos, consolidando esta información para elaborar un pedido al almacén central. Luego, el almacén revisa el *stock* disponible y prepara la solicitud a la droguería, considerando los análisis de máximos y mínimos de cada medicamento para garantizar el abastecimiento del policlínico.

Funcionalidades

Tomando en cuenta la información almacenada en la base de datos, el sistema debe de proveer resultados (tablas y gráficos) para cada una de las demandas descritas a continuación:

1. Listar los pacientes registrados junto a su número de identidad y dirección.
2. Mostrar la relación de doctores con su respectivo departamento, incluyendo nombre completo y cargo, junto a la diferencia del promedio que atiende mensualmente cuando tiene guardia con respecto a sus consultas por el departamento.
3. Listar todas las atenciones médicas realizadas a un paciente dentro un rango de fechas, mostrando la información del diagnóstico y el médico tratante, con el fin de dar continuidad al tratamiento.
4. Mostrar las últimas diez atenciones recibidas a un paciente, incluyendo médico, departamento y medicamentos recetados.
5. Consultar el consumo acumulado de un medicamento específico en todos los departamentos durante un mes, comparando los niveles disponibles con los máximos y mínimos definidos.
6. Presentar todas las solicitudes denegadas por el administrador del almacen principal, realizadas por los responsables de cada almacén dentro de los departamentos.
7. Determinar la tasa de éxito de prescripción de medicamentos para los pacientes mayores de 60 años con diagnósticos de enfermedades crónicas, comparando a los doctores por departamento. La "tasa de éxito" se define como el porcentaje de pacientes que, después de una prescripción inicial, no han requerido una nueva consulta por el mismo diagnóstico en los siguientes tres meses. La consulta debe mostrar los 5 doctores con la tasa de éxito más alta en este segmento de pacientes, junto con la cantidad total de prescripciones que han realizado para ellos, el departamento al que pertenecen y los medicamentos más frecuentemente prescritos por cada uno.

Además, la posibilidad de exportar la información mostrada a ficheros con formato PDF tiene que ser una funcionalidad provista por el sistema para todo tipo de usuario. Así como poder ordenar cada columna de los resultados acorde a los intereses del usuario final.

Requisitos particulares de la asignatura Ingeniería de Software

Además de la implementación de la aplicación web, las evaluaciones consistirán también de seminarios y preguntas escritas, los cuales se orientarán y se explicarán en su momento durante el semestre.

Con respecto al desarrollo de la solución, los requerimientos utilizados para calificar el trabajo serán:

1. Trabajar con un control de versiones (github, tfs, etc.).
2. Realizar la planificación con alguna herramienta CASE (github, jira, gantt, etc.).
3. Sistema multiplataforma.
4. Cumplir con todos los requerimientos funcionales planteados en el problema.
5. Tener buenas prácticas de programación, incluido los comentarios en todo el código (*docstring*).
6. Implementar al menos dos patrones.
7. Implementar una arquitectura que permita a la aplicación ser desacoplada, extensible en funcionalidades y mantenible .
8. Implementar pruebas unitarias, tanto como para el *back-end* como para el *front-end*.

No obstante, para considerar al equipo (estudiantes) aprobado en la asignatura es indispensable cumplir con los primeros seis puntos descritos.