

Proyecto #2 – HospiTEC

Instituto Tecnológico de Costa Rica
Escuela Ingeniería en Computadores
Bases de Datos (CE3101)
I Semestre 2024
Valor 20%



Objetivo general

- Desarrollar una aplicación que permita manejar la descripción del caso expuesto en el punto 4.

Objetivos específicos

- Aplicar los conceptos del modelo conceptual y relacional.
- Crear dos Base de Datos, una en PostgreSQL y otra en MongoDB.
- Crear un servicio API.
- Crear una página Web.
- Usar herramientas como Angular, Bootstrap, HTML5, CSS.
- Crear un documento de instalación.
- Crear un plan de proyecto.
- Crear al menos 2 diseños conceptuales para evaluar el análisis de problemas.

Descripción del problema

Con la última Pandemia del Covid-19 y el colapso que afectó al sistema médico nacional el Instituto Tecnológico ha decidido transformar su clínica de salud en el Hospital Tecnológico para que pueda atender cualquier emergencia nacional. Para la sostenibilidad financiera del Hospital se decidió que los servicios que ofrece el Hospital también puedan ser vendidos al público en general, lo que hace necesario la creación de un sistema administrador del hospital que se describe a continuación.

Vista Paciente: esta es la plataforma que permitirá a cualquier paciente crear una reservación para poder realizarse un procedimiento médico de los que se ofrecen en el hospital.

Además, permitirá observar el historial clínico y calificar tanto la estadía en el hospital como al personal que lo atendió.

Vista Personal Administrativo: esta es la plataforma que permite a los administradores del hospital ingresar la información general del sistema como: salones, camas, equipo médico, procedimientos, usuarios, etc.

Vista Doctor: esta plataforma permite al doctor: agregar pacientes, modificar su historial clínico y agregar medicamentos utilizados en el tratamiento clínico de cualquier patología.

Requerimientos del Software

- Vista Paciente.
 - ◆ **Crear Cuenta (Agregar Paciente):** el sistema debe permitir crear en el sistema a cualquier persona que así lo desee.
De un paciente se desea saber: nombre, apellidos, cédula, teléfono, dirección, fecha de nacimiento y patologías que tiene, así como el tratamiento que toma para esa patología.
 - ◆ **Gestión de Reservación:** el sistema debe permitir a un paciente crear, modificar y eliminar una reservación de una cama de hospital.

De una reservación se desea saber: el paciente, la fecha de ingreso, procedimientos médicos que se realizará y la fecha de salida (esta debe ser calculada y corresponderá a la sumatoria de la fecha de ingreso más los días requeridos por cada uno de los procedimientos médicos que comprende la reservación).

Una reservación puede realizarse si solo si el hospital tiene camas disponibles en las fechas indicadas por el paciente.

- ◆ **Ver Historial Clínico:** el sistema debe permitir a un paciente observar su historial clínico, así como los tratamientos otorgados. Debe ser listado cronológicamente de la fecha más reciente a la más antigua.
- ◆ **Evaluación del Servicio Recibido:** Considerando el aseo del hospital, trato del personal y puntualidad de las citas, esta evaluación **debe ser almacenada en MongoDB**.

→ Vista Doctor.

- ◆ **Agregar Paciente:** el sistema debe permitir a un médico crear en el sistema a cualquier persona que por alguna emergencia se presente al centro de salud y no esté registrado como paciente.
De un paciente se desea saber: nombre, apellidos, cédula, teléfono, dirección, fecha de nacimiento y patologías que tiene, así como el tratamiento que toma para esa patología.
- ◆ **Agregar Historial Clínico:** El sistema permitirá a los médicos agregar y modificar historia clínica a un paciente de la cual se desea saber: procedimiento realizado, fecha del procedimiento y tratamiento prescrito.

→ Vista Personal Administrativo.

- ◆ **Gestión de Salones:** El sistema debe proveer la posibilidad de crear, modificar y eliminar salones. Para el registro de salones se desea saber: número de salón, nombre del salón, capacidad en camas, si es medicina de mujeres, hombres o niños y piso en que se ubica.
- ◆ **Gestión de Equipo Médico:** El sistema permitirá la creación y modificación de equipo médico del cual se desea saber: nombre, proveedor, y cantidad disponible por default se conocen: luces quirúrgicas, ultrasonidos, esterilizadores, desfibriladores, monitores, respiradores artificiales y electrocardiógrafos.
- ◆ **Gestión de Camas:** El sistema debe proveer la creación y modificación de camas de las cuales se desea saber: el número de cama, equipo médico con que cuenta, salón en el que se encuentra y si es una cama la unidad de cuidados intensivos (UCI).
- ◆ **Gestión de Procedimientos Médicos:** El sistema debe proveer la creación y modificación de procedimientos médicos de los cuales se desea saber: el nombre y la cantidad de días mínimo de recuperación en salón. Inicialmente el hospital tendrá a disposición los siguientes procedimientos: Apendicectomía, Biopsia de mama, Cirugía de cataratas, Cesárea, Histerectomía, Cirugía para la lumbalgia, Mastectomía, Amigdalectomía.
- ◆ **Gestión Personal:** El sistema debe permitir crear, modificar y eliminar personal tanto administrativo como médico (doctores y enfermeros), de los cuales se desea saber: nombre, apellidos, cédula, teléfono, dirección, fecha de nacimiento y fecha de ingreso a la institución. Es importante poder realizar la diferencia entre personal administrativo, doctor o enfermero.
- ◆ **Carga de pacientes:** El sistema debe proveer la facilidad de realizar una carga inicial de pacientes que existen por default se proveerá un archivo XLSX (Excel) que deber ser importado a la base de datos. Se sugiere dejar los datos del archivo Excel en una tabla

temporal y por medio de un procedimiento almacenado se cargue a las tablas finales del sistema, este procedimiento almacenado será el encargado de realizar todas las validaciones necesarias.

- ◆ **Reporte áreas de mejora:** Tomado de las calificaciones de los pacientes. Debe identificar el aspecto evaluado y la nota recibida. **Se debe tomar de mongoDB.**

Requerimientos no funcionales del sistema

- El Sistema debe ser una aplicación web (utilizando Angular, Bootstrap, HTML5, CSS) y debe ser desplegada en la nube AWS o AZURE.
- La Base de Datos debe estar en PostgreSQL y MongoDB.
- La capa de servicios debe estar desarrollada en C# y debe ser desplegada en la nube AWS o AZURE.
- La Base de Datos debe estar al menos en tercera forma normal.
- Se deben implementar al menos 5 procedimientos almacenados, 2 funciones y 2 triggers.

Entregables

- Manual de Usuario.
- Documentación Técnica y del proyecto (descrita en el punto 6).
- Al menos 2 modelos conceptuales de la base de datos y el análisis de la elección del modelo final con respecto a los otros modelos.
- Documento de instalación.
- Plan de Proyecto.
- Script de Base de Datos.
- Script de población de Base de Datos.
- Aplicación WEB.
- Web API.
- Minutas.

Documentación requerida

- Se deberá documentar el código fuente.
- Se deberá entregar un documento que contenga:
 - ◆ Modelos conceptuales utilizando la notación de Chen.
 - ◆ Modelo relacional.
 - ◆ **Diagrama de arquitectura de la solución y su descripción.**
 - ◆ Descripción de las estructuras de datos desarrolladas (Tablas).
 - ◆ Descripción de los Store Procedures/Funciones/Triggers implementados.
 - ◆ Problemas conocidos: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.
 - ◆ Problemas encontrados: descripción detallada, intentos de solución sin éxito, soluciones encontradas con su descripción detallada, recomendaciones, conclusiones y bibliografía consultada para este problema específico.
- Documentación de evidencia del trabajo en equipo.
 - ◆ Actividades planeadas, su responsable y fecha de entrega. (Plan de trabajo)
 - ◆ Minutas de sesiones de trabajo. (Seguimiento al plan de trabajo)

- ◆ Actividades realizadas por cada estudiante. (Bitácora en digital, donde se describen las actividades realizadas, desde reuniones con el compañero de trabajo, investigaciones, consultas, etc. Se debe describir todo por más insignificante que sea, esto demostrará si ustedes están trabajando en realidad. Este es su diario de trabajo, llevan seguimiento de todo en el tiempo, imaginen que, si un compañero los releva en su trabajo, le bastaría con leer sus bitácoras para seguir el trabajo.
 - ◆ Evidencia de uso de un manejador de código (se recomienda Github).
- Conclusiones y Recomendaciones del proyecto.
- Bibliografía consultada en todo el proyecto

Aspectos operativos y evaluación:

1. **Fecha de entrega:** De acuerdo con el cronograma del curso y lo establecido en el TEC Digital. Se establece el siguiente plan de entregas parciales:
 - a. Plan de proyecto: 10 de Mayo 2024.
 - b. Resumen Ejecutivo Avance 1: 17/Mayo/2024
 - c. Resumen Ejecutivo Avance 2: 24/Mayo/2024
 - d. Funcionalidad completa: 2/Junio/2024
2. El proyecto tiene un valor de 20% de la nota del curso.
3. El trabajo es **en grupos de 5 personas**.
4. La implementación tendrá un valor de un 65% de la nota final, debe estar funcional. La defensa vale un 15% y la documentación un 20%.
5. Cumplir con los requerimientos especificados en la documentación no significa que se tienen todos los puntos, se evaluará que la documentación sea coherente, acorde al tamaño del proyecto y el trabajo realizado, no escatimen en documentación.
6. Cada grupo recibirá una nota en cada uno de los siguientes apartados Código y Documentación.
7. El profesor no sólo evaluará la funcionalidad del proyecto, esto quiere decir que aunque el proyecto este 100% funcional esto no implica una nota de un 100, ya que se evaluarán aspectos de calidad de código, aplicación del paradigma OOP, uso de herramientas solicitadas, calidad de documentación interna y externa, trabajo en equipo.
8. No se revisarán funcionalidades parciales, ni funcionalidades no integradas.
9. Es responsabilidad de cada miembro del grupo conocer su código, el profesor puede preguntar a cualquier miembro del grupo que le explique alguna funcionalidad/porción de código.
10. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
11. Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
 - a. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
 - b. Si no se utiliza un manejador de código se obtiene una nota de 0.
 - c. Si el código y la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
 - d. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
 - e. Si el grupo no cuenta con los equipos necesarios para realizar la revisión y no avisó al profesor de esta situación obtendrá una nota de 0.
 - f. El código debe ser desarrollado en C#, en caso contrario se obtendrá una nota de 0.

g. No presentarse a la revisión se obtiene nota de 0.

12. Cada grupo tendrá como máximo 60 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
13. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
14. Cada grupo es responsable de llevar los equipos requeridos para la revisión.
15. Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.
16. Las revisiones se realizan con los estudiantes matriculados en el curso, cualquier persona fuera de estos y los mencionados en el punto 15, no pueden participar en la revisión.
17. Si no se realiza la defensa del proyecto se asignará una nota de 0 en el proyecto.
18. Después de enviada la nota final del proyecto el estudiante tendrá un máximo de 3 días hábiles para presentar un reclamo siempre y cuando la funcionalidad esté completa.

Referencias

AngularJS (2018-10-04). Recuperado de: <https://angularjs.io>

Bootstrap Themes & Templates (2018-10-04). Recuperado de: <https://wrapbootstrap.com/>

How to Write Doc Comments for the Javadoc Tool. (2018-10-04). Recuperado de: <http://www.oracle.com/technetwork/articles/java/index-137868.html>

C# Coding Conventions (C# Programming Guide). (2018-10-04). Recuperado de: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>