Proyecto #2 - XTECDigital

Instituto Tecnológico de Costa Rica
Área Académica Ingeniería en Computadores
Bases de Datos (CE3101)
II Semestre 2020
Valor 25%

Objetivo general

- ➔ Desarrollar una aplicación que permita manejar la descripción del caso expuesto.

Objetivos específicos

- ➔ Aplicar los conceptos del modelo conceptual y relacional.
- ➔ Crear una Base de Datos relacional en Microsoft SQL Server para que permita el almacenamiento de los datos.
- ➔ Crear una Base de Datos **no relacional** en MongoDB para que permita el almacenamiento de los datos.
- ➔ Crear al menos 2 APIs (una para SQL Server y otra para MongoDB) para que controlen las funcionalidades y permitan la comunicación entre múltiples bases de datos.
- ➔ Crear una página Web para que exponga la funcionalidad al usuario.
- ➔ Usar herramientas como Angular, Bootstrap, HTML5, CSS, Entity Framework, y Reporting Services, Cristal Reports o similar para reportes.
- ➔ Crear un documento de instalación que permita el despliegue correcto de la aplicación en la nube.
- ➔ **Evaluar de forma objetiva, válida y precisa la solución planteada al problema complejo de ingeniería.**
- ➔ **Colaborar de forma activa en el equipo de trabajo para la realización del proyecto.**

Descripción del problema

La universidad XTEC ha decidido implementar un sistema que permita gestionar el proceso de enseñanza-aprendizaje.

Este sistema debe permitir gestionar todos los elementos relacionados con los cursos impartidos en XTEC, desde la perspectiva de los profesores y los estudiantes.

Dado lo anterior se ha determinado que son requeridas las siguientes vistas:

- **Vista Administrador:** Es la vista que usaran los administradores del sistema para aspectos de configuración de los cursos impartidos en cada semestre.
- **Vista Profesor:** esta es la plataforma que permitirá a los profesores poder gestionar todos los componentes relacionados a sus cursos.
- **Vista Estudiante:** Ésta es la plataforma que permite a los estudiantes gestionar los cursos matriculados en cada semestre.

Requerimientos del Software

- ➔ Vista Administrador.
  - ◆ **Log In:** Se debe contar con una cuenta de administrador que permita autenticarse en esta vista. Incluirlo en el script de población de la base de datos de información personal de profesores y estudiantes (ver información de esto más adelante).
  - ◆ **~~Gestión de Cursos~~:** Debe permitir gestionar (visualizar, crear o deshabilitar) la lista de cursos genérica que utiliza el sistema. Para cada curso se requiere el código del curso, nombre del curso, cantidad de créditos y la carrera a la que pertenece.
  - ◆ **~~Inicializar Semestre:~~** A continuación, se describe el proceso para inicializar un semestre:
    - Debe permitir crear semestres. Para cada semestre debe indicarse el año y el periodo (1 para el primer semestre, 2 para el segundo semestre y V para el periodo de verano).
    - Luego debe establecerse los cursos que serán impartidos en el semestre. Para cada curso debe indicarse el número de grupo y el profesor del curso. Existen casos donde un grupo está asignado a dos profesores.
    - Para cada grupo debe establecerse los estudiantes matriculados. Esto debe realizarse mediante el número de carné. La información personal de cada estudiante debe obtenerse de otra base de datos (más adelante se brindará el detalle de esto).
    - Cada grupo tiene asociado una sección de documentos y la parte de evaluaciones.
    - La sección de documentos debe crearse inicialmente con la siguiente estructura de carpetas:
      - Presentaciones
      - Quices
      - Exámenes
      - Proyectos
    - La sección de evaluaciones debe crearse inicialmente con los siguientes rubros:
      - Quices (30%)
      - Exámenes (30%)
      - Proyectos (40%)
    - El proceso de Inicialización de un semestre puede hacerse de forma manual, paso por paso o también puede hacerse mediante la carga de un archivo de excel que tenga toda la información, la cual debe cargarse en una tabla temporal y debe ejecutarse un **Procedimiento Almacenado** que lea la información de la tabla temporal y ejecute el proceso descrito anteriormente.

➔ Vista Profesor.

- **Log In:** Se debe contar con una cuenta de profesor que permita autenticarse en esta vista.
- **Gestión de Documentos**: El profesor puede visualizar, editar, agregar o eliminar documentos en los cursos. Al agregar un documento el profesor debe seleccionar un archivo de su computador y seleccionar la opción de cargar el archivo. Una vez que se ha subido el archivo debe visualizarse la fecha en la que lo subió, el tamaño del archivo y permitir visualizar el archivo.
  El profesor puede utilizar la estructura de carpetas definida en la creación de cada curso o puede crear carpetas. Puede eliminar carpetas pero solo las que haya creado el profesor, es decir, las carpetas creadas inicialmente para el curso no pueden eliminarse (Presentaciones, Quices, Exámenes, Proyectos).
- **Gestión de Rubros:** Se puede visualizar, editar, agregar o eliminar rubros. Si se permite cambiar los rubros establecidos para cada curso en el proceso de Inicializar Semestre. Para cada rubro es necesario un nombre y un porcentaje total del rubro. Se debe validar que la suma de los rubros sea 100.
- **Asignar Evaluaciones:** El profesor asignará evaluaciones. Para las evaluaciones se debe establecer el rubro al que pertenece, el peso de la evaluación con respecto al rubro seleccionado, fecha y hora máxima de entrega, subir la especificación de la evaluación y definir si la evaluación es individual o grupal. En caso de establecerse que la evaluación es grupal debe permitir definir la conformación de los grupos.
- **Evaluar entregables**: El profesor debe poder descargar los entregables de cada evaluación para así poder asignar una nota a cada entregable. También debe permitir agregar observaciones o subir un archivo con el detalle de la nota obtenida.
  Cuando el profesor pone las notas puede guardar las notas o publicarlas. En caso de optar por guardarlas, las notas se almacenan en el sistema pero solo el profesor puede verlas. Hasta que el profesor seleccione la opción de publicar es cuando los estudiantes pueden visualizarlas.
  Cuando el profesor seleccionar la opción de publicar se debe implementar un ~~**Trigger**~~ que generé de forma automática una noticia indicando a los estudiantes que pueden revisar las notas de la evaluación realizada por el profesor.
- **Gestión de Noticias**: El sistema debe permitir visualizar, crear, modificar y eliminar noticias. Para una noticia se requiere un título, un mensaje, una fecha de publicación y el autor de la noticia. Debe estar asociada a un curso.
- **Reporte de notas:** Debe permitir visualizar un reporte de notas de todos los estudiantes que forman parte del grupo del curso. Para cada estudiante debe mostrar el detalle de todas las notas y calcular el valor obtenido para cada rubro, así como la nota final curso. Este reporte debe implementarse utilizando una **Vista en la base de datos** y usando herramientas como Reporting Services o Cristal Reports la idea que se busca es un archivo PDF que pueda ser impreso.
- **Reporte de Estudiantes Matriculados:** Mostrar la lista de estudiantes matriculados en el curso, indicando carnet, **nombre, correo electrónico y teléfono**. Este reporte debe presentarse utilizando una **Vista en la base de datos**

y usando herramientas como Reporting Services o Cristal Reports. Debe permitir descagar la información en un PDF.

- ➔ Vista Estudiante.
  - **Log In:** Se debe contar con una cuenta de profesor que permita autenticarse en esta vista.
  - **Visualizar Documentos:** Debe permirir a los estudiantes visualizar la estructura de documentos establecida por el profesor. Debe permitir navegar por esta estructura y acceder a los documentos que ha publicado el profesor. Puede descargar cualquiera de estos archivos.
  - **Enviar Evaluaciones:** Para las evaluaciones que ha asignado el profesor, los estudiantes deben tener la posibilidad de cargar el archivo correspondiente al entregable. Si la entrega es grupal y algún estudiante del grupo ya lo subió se debe indicar que ya se envió. Se debe permitir descargar el entregable que fue previamente cargado en la herramienta.
  - **Reporte Notas del curso:** Mostraria la misma información que el reporte de notas que visualiza el profesor, con la única diferencia que el estudiante solo vería la información de la fila correspondiente a sus notas.
  - **Visualizar Noticias:** Mostrar una lista de noticias publicadas por el profesor, **ordenadas de forma descendente** por la fecha de publicación. La noticia más reciente de primero.

Para la información personal de estudiantes y profesores debe implementarse una base de datos no relacional en **~~MongoDB.~~** Para esta base de datos debe considerar lo siguiente:

- ➔ Para los estudiantes se requiere la siguiente información:
  - Carné
  - Nombre
  - Correo electrónico
  - Teléfono
  - Password (encriptado usando MD5)
- ➔ Para los profesores se requiere la siguiente información:
  - Cédula
  - Nombre
  - Correo electrónico
  - Password (encriptado usando MD5)

La base de datos principal de XTECDigital no almacena información personal de estudiantes, profesores o administradores. La información que deba almacenarse en esta base de datos con relación a estudiantes, profesores y administradores debe usar el número de carné y cédula respectivamente. Cualquier información personal debe ser consultada de la base de datos no relacional. Para el acceso a ambas bases de datos debe realizarse mediante un API.

Para la revisión del proyecto:

- ➔ Aparte del procedimiento almacenado solicitado en la especificación, el profesor seleccionará tres adicionales para ser evaluados. En total se evaluarán cuatro procedimientos almacenados (el obligatorio y tres adicionales seleccionados durante la revisión).
- ➔ Aparte del trigger solicitado en la especificación, el profesor seleccionará dos adicionales para ser evaluados. En total se evaluarán tres triggers (el obligatorio y dos adicionales seleccionados durante la revisión).
- ➔ Aparte de la vista solicitada en la especificación, el profesor seleccionará dos adicionales para ser evaluadas. En total se evaluarán tres vistas (la obligatoria y dos adicionales seleccionadas durante la revisión).

Requerimientos no funcionales del sistema

- ➔ El Sistema debe ser una aplicación web (utilizando Angular, Bootstrap, HTML5, CSS).
- ➔ Las Bases de Datos deben estar en Microsoft SQL Server y MongoDB.
- ➔ **Toda la lógica de la Base de Datos debe ser implementada mediante Store Procedures, Vistas y Triggers.**
- ➔ La capa de servicios debe estar desarrollada en C# y debe ser desplegada en **Azure o AWS** al igual que la App web.
- ➔ El equipo de trabajo debe seleccionar a uno de sus miembros como único punto de contacto. Todas las comunicaciones y solicitudes deben ser a través de dicho punto de contacto.

Entregables

- ➔ Manual de Usuario.
- ➔ Documento de evidencia de la solución elaborada y planificación del trabajo.
- ➔ Se deberá documentar el código fuente.
- ➔ Evidencia de uso de un manejador de código (se recomienda Gitgub).
- ➔ Documento de instalación.
- ➔ Plan de Proyecto.
- ➔ Script de Base de Datos.
- ➔ Script de población de Base de Datos.
- ➔ Aplicación WEB.
- ➔ Web APIs.
- ➔ Minutas.

NOTA: Cada documento solicitado debe tener la estructura de un documento técnico (portada, índice de contenidos, introducción, entre otros).

Documentación evidencia de solución elaborada y planificación del trabajo

- ➔ Se deberá entregar un documento que contenga:
  - ◆ Modelo conceptual utilizando la notación de Chen.
  - ◆ Modelo relacional.
  - ◆ Descripción de las estructuras de datos desarrolladas (Tablas).
  - ◆ Descripción de los Store Procedures, Triggers y Vistas implemendos.
  - ◆ **Descripción detallada de la arquitectura desarrollada.**
  - ◆ 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 (plan de proyecto).
  - ◆ Descripción de roles.
  - ◆ Actividades planeadas, su responsable, estimación de completitud de la tarea (menores a 8hs) y fecha de entrega. (Plan de trabajo)
  - ◆ Minutas de sesiones de trabajo. (Seguimiento al plan de trabajo)
  - ◆ Actividades realizadas por cada estudiante. (Bitácora, donde se describen las actividades realizadas por cada miembro del equipo de trabajo, de manera que se evidencie la participación en el proyecto de todos los estudiantes. La bitácora debe incluir desde reuniones con compañeros de trabajo, investigaciones, consultas, entre otros. Se debe describir todo por más insignificante que sea, esto demostrará si ustedes están trabajando en realidad. Este es su diario de trabajo.
- ➔ Conclusiones del proyecto.
- ➔ 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: 25/Nov/2020
   b. Resumen Ejecutivo Avance 1: 2/Dic/2020
   c. Resumen Ejecutivo Avance 2: 9/Dic/2020
   d. Funcionalidad completa: 18/Dic/2020
2. El proyecto tiene un valor de 25% de la nota del curso.
3. El trabajo es **en grupos de 4 personas**.
4. La implementación tendrá un valor de un 70% de la nota final, debe estar funcional. La defensa vale un 10% y la documentación externa 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, Defensa 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 cero en el proyecto.
    b. Si no se utiliza un manejador de código se obtiene una nota de cero en el proyecto.
    c. Si el código y la documentación no se entregan en la fecha indicada se obtiene una nota de cero en el proyecto.
    d. Si el código no compila se obtendrá una nota de cero en el proyecto, 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 cero en el proyecto.
    f. El código debe ser desarrollado en C#, en caso contrario se obtendrá una nota de cero en el proyecto.
    g. Si el grupo no se presenta a la revisión/defensa se obtiene nota de cero en el proyecto. Si un estudiante no se presenta a la defensa y no cuenta con una justificación válida se le asignará al estudiante una nota de cero en el proyecto.
12. Cada grupo tendrá como máximo 50 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 y el profesor dispondrá de 10 minutos para dar la retroalimentación y la nota del proyecto.
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.

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