

Ingeniería en Computación
Bases de Datos 1 – Periódico Digital
Profesora: Adriana Álvarez

1. Motivación y problema

Los periódicos digitales han tomado relevancia en los últimos años dado el involucramiento de la comunidad en aspectos importantes de su día a día, el acceso a internet y a herramientas tecnológicas.

Las universidades públicas han querido publicar el quehacer de su organización, los trabajos científicos y de acción social en los cuales ha estado trabajando y sus resultados.

Sin embargo, los periódicos con mayor difusión y popularidad no están enfocados sólo en noticias universitarias, sino en muchos otros tópicos a nivel nacional con incluso mayor rating.

Con el propósito de divulgar el trabajo que se realiza en las universidades públicas, el aporte social, económico y científico que aportan con su trabajo y mostrar transparencia y rendición de cuentas con los fondos públicos que se manejan, es que a usted lo contratan para crear un sistema tipo periódico digital.

Ese sistema será un periódico donde se publiquen noticias relacionadas con el trabajo que está haciendo la comunidad universitaria, pero además debe recopilar información de los que leen las noticias a efectos de conocer el impacto en la difusión del trabajo que las universidades realizan.

Estructure su aplicación como un Sistema NCapas. Investigue más detalles sobre esta arquitectura, de modo que cuente con al menos los componentes básicos ilustrados en la Figura 1. A continuación se da una descripción general de cada una de las capas:

- *User Interface (UI)*: capa de presentación, encargada de mostrar las vistas de interfaz gráfica mediante la cual los usuarios interactúan con la aplicación. Esta capa no toma decisiones ni procesa lógica de negocio, solamente se encarga de mostrar o capturar información.
- *Business Layer (BL)*: capa de lógica de negocio, encargada de modelar las reglas de negocio de la aplicación, toma las decisiones, cálculos, procesos, etc.
- *Data Access (DA)*: capa de acceso a datos, abstrae la comunicación del sistema con la base de datos, no debe confundirse con la base de datos como tal.

Arquitectura

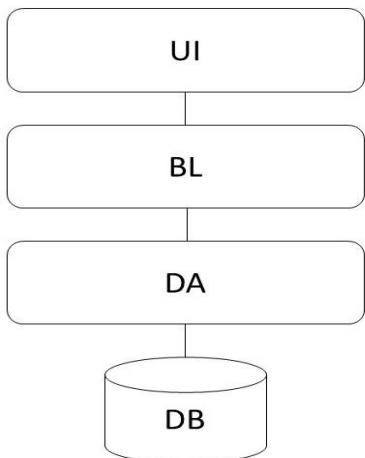


Figura 1 – Arquitectura N-Capas

Es probable que durante el desarrollo del sistema requiera de la implementación de capas adicionales. Durante la revisión se deberá justificar las decisiones hechas sobre el diseño de la aplicación. Adicional a esto se espera una implementación elegante, con un código fuente que siga las normas básicas establecidas en el libro Clean Code (Martin, 2008).

2. Requerimientos generales

- 1) Registro de personas
 - a) En este registro usted guardará a las personas que utilicen el sistema.
 - b) El sistema puede permitir que usuarios no identificados lean las noticias, pero para comentar o calificar las noticias es requisito crear un usuario.
 - c) El registro de personas debe incluir al menos los siguientes datos:
 - (1) Nombre
 - (2) Cédula
 - (3) Tipo de identificación con máscaras
 - (4) Género (femenino, masculino, otro)
 - (5) Teléfonos
 - (6) Correo electrónicos
 - (7) Dirección
 - (8) Foto
 - (9) Otros que usted considere necesarios
- 2) Funcionalidad del sistema:
 - a) El sistema debe permitir crear noticias por parte de autores y el tipo de autor debe estar claramente definido. De forma tal que se tendrán autores profesores, administrativos, estudiantes.
 - b) No todos los usuarios pueden crear noticias, sólo los definidos como autores.
 - c) La comunidad universitaria estará conformada incluso por vecinos de la universidad, por lo que ellos también podrían publicar noticias relacionadas con el vecindario.
 - d) El sistema debe permitir clasificar las noticias en:
 - i) Investigaciones científicas
 - ii) Trabajo social
 - iii) Política
 - iv) Cultura
 - v) Deporte
 - vi) Sucesos
 - vii) Opinión
 - viii) Ofertas de trabajo
 - ix) Eventos
 - (1) Sociales tales como fiestas, juegos como Random, otros
 - (2) Ambientales tales ventas de artículos pro-ambiente.

- Intercambio de bienes, entre otros.
- (3) Deportivos tales como yoga, spinning, otros.
- x) Otros.
- e) Existe un comité que aprueba los artículos que serán publicados en el periódico digital.
- f) Los artículos tienen al menos las siguientes características:
- i) Autor
 - ii) Título
 - iii) Texto de la nota
 - iv) Foto
 - v) Entre otros que usted considere
- 3) Edición y actualización: El sistema permite editar y actualizar los datos de todo el sistema según permisos y seguridad.

3. Canje de productos

Para incentivar a la comunidad a publicar sus artículos, los autores podrán comprar productos de un catálogo de productos por medio de puntos.

Al alcanzar 15 favoritos, los autores podrán comprar productos del catálogo.

Cada producto tiene un precio equivalente a favoritos. Por ejemplo, una pizarra vale 10 puntos que equivalen a 10 puntos de favoritos.

El sistema debe llevar el control de los puntos ganados, canjeados y pendientes por cada autor.

4. Manejo y consulta de bitácora

Cualquier cambio del texto de una noticia que ya fue publicada debe registrarse en una bitácora de sistema con la fecha, hora, descripción del cambio, texto anterior y texto actual y el nombre del usuario que hizo la operación.

Utilice triggers para cumplir con esta funcionalidad. Adicionalmente, en este módulo debe incluir una consulta a la bitácora filtrable por rango de fecha, hora, texto del cambio anterior y nuevo y usuario.

También el sistema debe cumplir con campos de auditoría que incluyan usuario que crea el registro, fecha de creación, último usuario que modifica un registro, última fecha de modificación.

5. Control y Usabilidad

El sistema permite a los administradores las operaciones de agregar, editar, eliminar o inactivar registros de los catálogos.

Todos los catálogos deben registrarse, editarse y consultarse por medio del sistema. Se considera no aceptable el registro de datos por medio de la base de datos.

6. Comentarios y Reviews

- a) El sistema debe permitir comentarios a los usuarios lectores registrados sobre los artículos publicados.
- b) Los artículos publicados pueden tener un review de los usuarios lectores. Este review se visualiza en el sistema con cantidad de estrellas dadas. De forma numérica y visual en estrellas y se debe

resaltar las estrellas obtenidas del total de 5 estrellas.

7. Favoritos

- a) Los usuarios pueden guardar sus noticias favoritas.

5) Flexibilidad, escalabilidad y parametrización

- a) El sistema debe ser flexible.
- b) El sistema debe permitir escalabilidad:
 - i) El sistema debe servir para implementar en otras universidades almacenando la información en la misma base de datos
 - ii) Los datos de una universidad no pueden ser vistos por otra universidad
 - iii) Se debe guardar la ubicación geográfica de la sede central de cada universidad
- c) El sistema debe ser completamente parametrizable:
 - i) No se permiten datos en duro en el código
 - ii) Todos los parámetros deben ser modificados desde la aplicación por los usuarios administradores según corresponda en seguridad

6) Seguridad

- a) El *password* debe encriptarse con un algoritmo de encriptación reconocido tales como RSA, AES, SHA, entre otros.

- b) La clave debe quedar encriptada en la base de datos y desencriptarse a la hora de verificar el login.
- c) Debe usar una librería pública y hacer la referencia en la bibliografía.
- d) Puede encriptar otros campos que considere sensibles del proyecto.

8. Pantalla principal

La pantalla principal de sistema debe mostrar lo siguiente en diferentes segmentos pero dentro de la misma pantalla:

1. Últimas noticias.
2. Noticias más vistas.
3. Debe mostrar 1, 2 o 3 noticias de cada clasificación.
4. Las noticias se muestran siempre con foto.

9. Otras pantallas

El sistema debe mostrar un menú por cada tipo de calificación de noticias.

Debe tener un menú administrativo para catálogos y temas administrativos (parámetros, por ejemplo) visible sólo a los administradores.

10. Consultas

Las consultas que debe ofrecer el sistema son:

1. Se deben mostrar todas las noticias (autor, título, fecha de publicación, tipo de artículo) del periódico ordenados

por antigüedad del más reciente al más antiguo.

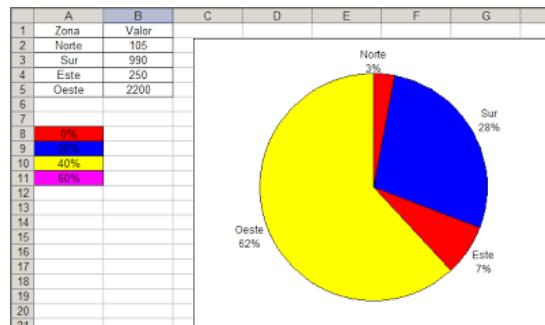
2. Listado de noticias por tipo de noticia.
3. Listado de favoritos para el usuario conectado por antigüedad desde la más reciente a la más antigua.
4. Listado de productos, sus valores y disponibilidad.
5. Listado de puntos ganados, canjeados y pendientes por autor.
6. Aspectos generales de las consultas:

- a. Los administradores y los autores pueden ver todas las noticias, puntos, productos.
- b. Filtros generales: autor, tipo de noticia, fecha.
- c. El listado de noticias debe tener un link al artículo completo, a sus reviews y comentarios.
- d. Cada consulta debe incluir el dato del total de registros consultados.

11. Estadísticas

Este módulo les permitirá a los administradores tener información precisa e inmediata de su información.

Cada una de las estadísticas deben mostrarse con datos numéricos, porcentajes y gráficos. Por ejemplo:



Las estadísticas mínimas que debe ofrecer el sistema son:

- a. Total de artículos publicados por universidad.
- b. Total de autores por universidad.
- c. Total de artículos publicados por autor ordenado de forma descendente.
- d. Top n de autores que más publican artículos (mostrando la cantidad de artículos publicados). Filtros: universidad, género, tipo de noticia.
- e. Promedio de la calificación del review por autor ordenado de forma descendente.
- f. Total de autores por rango de edad. De 0 a 18, de 19 a 30, de 31 a 45, de 46 a 60, de 61 a 75 y mayor de 75 por género y con filtro género. Filtros: universidad, género, tipo de noticia.
- g. Productos más canjeados indicando la cantidad de canjes.

12. Jobs

El sistema debe listar las noticias más gustadas de toda la historia del periódico digital con la noticia y la cantidad de favoritos. Esto se debe realizar por medio de un job que se ejecuta automáticamente todos los días a la 1 a.m.

13. Otras consideraciones

1. La aplicación debe guardar las fotos de los usuarios y de los artículos. Se recomienda en formato Base64.
 2. Debe utilizar Oracle como base de datos.
 3. Usted debe identificar donde aplica crear tablas catálogo o de restricción de dominio para que el sistema sea flexible.
 4. El sistema debe cumplir con los atributos de calidad de usabilidad, flexibilidad, fácil modificación, escalabilidad, seguridad, entre otros.
 5. La aplicación debe ser gráfica.
 6. La aplicación debe ser de escritorio.
 7. La aplicación debe tener registrados y probados los datos necesarios para las pruebas durante la defensa. Al menos lo siguiente:
 - a. 20 personas
 - b. Ubicación geográfica de Costa Rica
 - c. 15 noticias
 - d. 3 universidades
 - e. 10 reviews y comentarios
 - f. 15 favoritos
 - g. 10 productos
 - h. Todos los catálogos de restricción de dominio deben estar completamente registrados en el sistema
 - i. Otros que usted considere necesarios
8. Se evaluará el diseño y la correcta identificación de los parámetros.
 9. La aplicación debe tener un logo creado por el equipo. No puede ser un logo copiado de internet o de otras personas y debe ser original.
 10. El código debe trabajarse desde un programa de control de versiones y se deberá demostrar en la defensa la cantidad de commits realizados por cada miembro del equipo. Se recomienda Gitlab.
 11. El equipo puede estar conformado por grupos de un número máximo según le indique el profesor.
 12. La entrega oficial se debe hacer en un zip en el Tec Digital antes de la fecha y hora ahí indicada. No se aceptarán trabajos posteriores a esa fecha y hora. En caso de problemas de conexión, se debe aplicar el plan B de la presentación de Reglas del Juego.
 13. La entrega debe contener los archivos fuente y el ejecutable de la aplicación (si aplica), no está permitido subir links que dirijan a los documentos o carpetas.
 14. Cualquier sospecha de copia anulará el trabajo y se procederá con el trámite administrativo.
 15. El mejor proyecto tendrá +1 punto extra en la nota final del curso.

14. Documentación

La siguiente documentación debe ser entregada:

1. Diagrama de entidad – relación (modelo conceptual).
2. Manual de usuario con la descripción del propósito y uso del sistema que incluya imágenes significativas de la aplicación. Si utiliza un wiki para la documentación, esta debe accederse desde la aplicación.
3. Diccionario de datos.
4. Documento de matriz de casos de prueba (en Excel) con al menos 50 casos de prueba

con los cuales se probará el programa.

Debe contener un id como enumerador del caso de prueba, la descripción del caso de prueba, el resultado esperado y el estado (Certificado, Con Error, Pendiente). Además de una tabla pivote con la cantidad de casos de prueba en cada estado y el total.

5. El código debe estar documentado internamente según Clean Code.
6. Los procedimientos, funciones o paquetes importantes deben ir documentados con descripción, el autor principal, fecha de creación.