

Ingeniería en Computación
Bases de Datos 1 – Propuestas Ciudadanas
Profesora: Adriana Álvarez

1. Motivación

La participación ciudadana en las comunidades siempre ha sido un pilar para que estas tengan un desarrollo correcto.

Hoy en día las personas tienen más conciencia acerca de las problemáticas ambientales, sociales y comunales.

Proponen ideas que deben ser escuchadas y que prometen mejorar muchas de las situaciones actuales que afectan a la mayoría.

2. El problema

Lamentablemente en la actualidad las municipalidades no conocen en su totalidad las problemáticas de las comunidades.

Se cuenta con herramientas tales como las redes sociales para hacer aportar ideas y mejoras para todos las cuales han tenido gran uso y aceptación de parte de los ciudadanos por su facilidad. Sin embargo, esas herramientas no permiten que se desarrollen como proyectos formales que puedan ser escogidos por los mismos ciudadanos.

3. Arquitectura

Estructure su aplicación como un Sistema N Capas. 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.

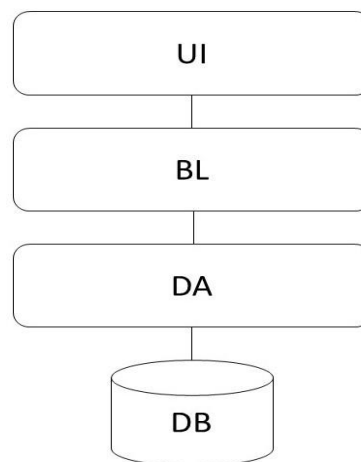


Figura 1 – Arquitectura N-Capas

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).

4. *Requerimientos*

a. Registro

El sistema debe permitir lo siguiente:

- a) Administradores para las operaciones de seguridad y administración.
- b) Registro de personas que ingresan al sistema con al menos nombre, apellidos, cédula o su equivalente en pasaporte, fecha de nacimiento, edad, foto, nacionalidad, comunidad a la que pertenecen, correo electrónico y teléfono.
- c) Cada persona que se conecta al sistema debe hacerlo por medio de un usuario y clave. La clave debe almacenarse encriptada.
- d) Deben existir usuarios Administradores para registrar la información de catálogos.
- e) Todos los datos pueden ser editados considerando los permisos de seguridad.
- f) Las personas pueden registrar propuestas en general para su comunidad, la cuales cuentan con un título, una descripción, categoría y un presupuesto aproximado.
- g) También los usuarios podrían votar por propuestas de otras

personas para que reciban apoyo y se lleguen a ejecutar.

- h) Existirá un muro de comunidad en el cual los ciudadanos podrán colgar comentarios y discutir acerca de las necesidades de cada una de las mismas, cada ciudadano tendrá acceso al muro respectivo registrado en su usuario, por ejemplo, Jose tiene su usuario registrado en Guadalupe, por lo tanto, solamente tendrá acceso a propuestas y al muro de discusión de Guadalupe)

b. Registro de propuestas

El sistema debe permitir lo siguiente:

- a) Clasificar los tipos de propuestas. Por ejemplo, ambientales, recreativas, de seguridad, etc.
- b) Registro de nombre de la propuesta, descripción, fecha, presupuesto aproximado y clasificación.
- c) Ciudadano: Se debe registrar la información del expositor de la propuesta, tal como nombre, y correo de contacto.

c. Publicación de mensajes en el foro de discusión.

- a) Debe aparecer el nombre de la persona que publica el comentario.
- b) Hora a la que se realiza el comentario.
- c) Texto del comentario.

d. Favoritos

El sistema debe permitir a los usuarios optar por las categorías favoritas, de forma tal que se le muestren con

prioridad y de primero las propuestas de las respectivas categorías que marque como favoritas.

Este punto no limita que pueda visualizar el resto de la información.

e. Auditoría y bitácora

El sistema debe administrar campos de auditoría para los registros de datos en las tablas de la base de datos y para la última actualización de un registro.

Además, debe tener una bitácora de cambios de clave de un usuario.

Utilice *triggers* para cumplir con esta funcionalidad donde se requiera.

Adicionalmente, en este módulo debe incluir una consulta de los usuarios que modificaron su clave a partir un rango de fechas seleccionable por el administrador.

f. Control y Usabilidad

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

g. Módulo de Consultas

Este módulo les permitirá a los usuarios visualizar información precisa e inmediata.

Todas las consultas deben contener filtros de los datos que se solicitan.

Todas las consultas deben permitir combinación de datos en los filtros y filtros en blanco.

Las consultas mínimas que debe ofrecer el sistema son:

Consultas para administradores:

- Listado de los usuarios cuyas claves no han sido modificadas en los últimos 10 días incluyendo la cantidad de días. El reporte debe incluir un total de personas al final. Filtros mínimos: nombres, apellidos, cédula, usuario.
- Top "n"(valor parametrizable) de comunidades que más propuestas tienen.

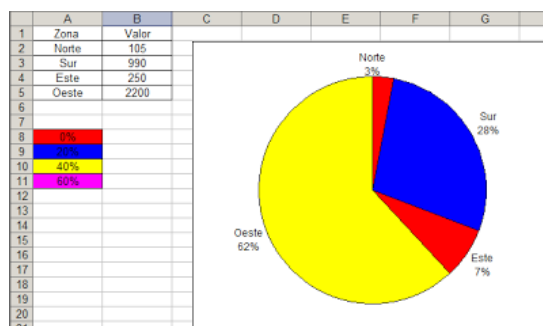
Para usuarios:

- Listado de propuestas por clasificación. Debe incluir totales. Filtros mínimos: clasificación, rangos de fecha, cantidad de votos.
- Top "n"(valor parametrizable) de propuestas con más votos.
- Listado de las propuestas realizadas por ellos.

h. Módulo de 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:

Todas las estadísticas deben permitir combinación de datos en los filtros.

- a) Total y porcentaje de propuestas por clasificación.
- b) Total y porcentaje de propuestas por zona, Filtros mínimos: país, provincia, cantón, distrito, comunidad.
- c) Rango de edad y porcentaje de usuarios. De 0 a 18, de 19 a 30, de 30 a 45, de 46 a 55, de 55 a 65, de 66 a 75, de 76 a 85, mayores a 85.
- d) Total y porcentaje de usuarios por lugar donde viven. Filtros: país, provincia, cantón, distrito, comunidad.

i. Jobs

El sistema debe generar un reporte al día con la propuesta más votada de **todas las comunidades**.

Este reporte debe generarse como un Job de base de datos que corre todos los días a las 11 p.m. de lunes a domingo.

5 Otras consideraciones

1. El sistema debe presentarse en la defensa con datos reales y bien clasificados.
2. Los catálogos deben tener los datos previamente registrados para la defensa del proyecto. Al menos 20 propuestas, 30 comentarios de discusión, toda la información de catálogos.
3. Todos los campos que se ingresen en el sistema deben tener su respectiva salida (consulta,

estadística, reporte, otro que considere).

4. Debe utilizar Oracle como motor de base de datos.
5. Usted debe identificar donde aplica crear tablas catálogo para que el sistema sea flexible. Se calificará diseño.
6. El sistema debe cumplir con los atributos de calidad de usabilidad, flexibilidad, fácil modificación, entre otros.
7. La aplicación debe ser gráfica.
8. La aplicación debe ser de escritorio.
9. Se evaluará el diseño y la correcta identificación de los parámetros.
10. La aplicación debe tener un nombre y logo creado por el equipo. No puede ser un logo copiado de internet o de otras personas y debe ser original. Puede solicitar ayuda de un estudiante de diseño.
11. 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 por ejemplo Gitlab, Github.
12. El equipo puede estar conformado por grupos de un número máximo según le indique el profesor.
13. La entrega oficial se debe hacer en un zip en el tecDigital 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.

14. La entrega debe contener los archivos fuente y el ejecutable de la aplicación (si aplica).
15. La documentación debe entregarse en pdf, latex, doc. No se acepta un link al documento.
16. Cualquier sospecha de copia anulará el trabajo y se procederá con el trámite administrativo.
17. El mejor proyecto tendrá +1 punto extra.

6 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 y el wiki completo debe ser entregado en el tecDigital.

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. Los procedimientos, funciones o paquetes importantes deben ir documentados con descripción, el autor principal, fecha de creación.

7 Referencias

<https://decide.montesdeoca.go.cr/>