

Ingeniería en Computación
Administración de Proyectos – Bandera Azul
Profesora: Adriana Álvarez

1. Motivación y problema

El consumo de productos con envases ha incrementado considerablemente en las últimas décadas.

La industria y los negocios fomentan el uso de materiales no reciclables o un exceso de recipientes.

Por ejemplo, las empresas proveedoras de servicios alimentarios usan materiales no reciclables como el estereofón.

Otras generan un exceso de papel, cartón, plástico, vidrio y lamentablemente no se tiene en el país la cultura de la clasificación de residuos que permita reutilizar materiales reciclables.

Los productos que se compran en el supermercado generan gran cantidad de envoltorios, paquetes, bolsas, entre otros que se botan en la basura y van a depositarse en los rellenos sanitarios o en el mar.

Esta contaminación afecta animales terrestres y marinos.

Se han visto casos tristes de tortugas deformadas por enredarse en bolsas plásticas, tortugas con pajillas incrustadas en sus narices, ballenas muertas por consumir plásticos tirados al mar, animales ahogados en bolsas plásticas.

Se han encontrado en el mar islas de plástico del tamaño de países continentales.

El propósito de este proyecto es estimular en la población costarricense la

correcta clasificación de los residuos valorizables con el objetivo de disminuir los problemas que causa el excesivo uso de recipientes y su mala gestión.

El desarrollo del proyecto será la creación de un sistema que premie a los ciudadanos por realizar una correcta clasificación, limpieza y recolección de materiales reutilizables.

El sistema se basará en el último plan nacional de reciclaje.

2. Arquitectura

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.

- *Capa de Integración de Servicios Externos:* Para manejar conexiones con APIs externas, como servicios de mapas (por ejemplo, Google Maps o OpenStreetMap) para rutas.
- *Capa de Seguridad:* Encargada de la autenticación y autorización y manejo seguro de sesiones.
- *Capa de Caché y Optimización:* Para almacenar respuestas de consultas frecuentes y mejorar el rendimiento de la aplicación (opcional).

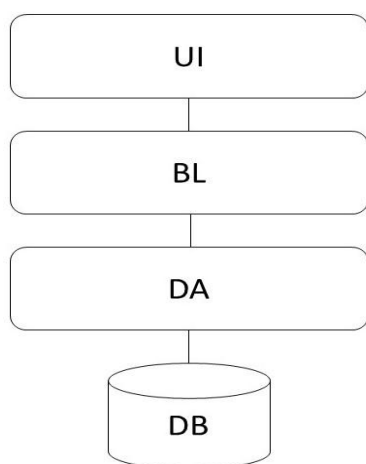


Figura 1 – Arquitectura N-Capas

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).

3. Justificación del Diseño

La elección de una arquitectura en capas permite una clara separación de responsabilidades, lo que facilita el mantenimiento del código y la realización de pruebas de manera más efectiva. Se deben utilizar los principios de "Clean Code" de Robert C. Martin (2008), promoviendo un código limpio, legible y fácil de mantener.

Este enfoque proporciona una base sólida para la escalabilidad del sistema, ya que cada capa puede ser mejorada o modificada de forma independiente, permitiendo que la aplicación evolucione según las necesidades de los usuarios y el mercado.

4. Requerimientos

4.1 Registro de usuarios.

El sistema debe permitir el registro de usuarios que están interesados en aplicar al sistema de clasificación de materiales reutilizables y ganar puntos por ello. Así como administradores super usuarios, centros de acopio y comercios afiliados.

- a) Registro de Usuarios y Gestión de perfiles: El sistema debe permitir el registro de usuarios (estudiantes y/o funcionarios) con autenticación basada en credenciales. Los usuarios deben poder crear y gestionar un perfil, incluyendo información básica como nombre, correo, teléfono, tipo de identificación, cédula, fecha de nacimiento, edad, género, fotos. Cuando el usuario se registra, debe poder leer y

aceptar los términos y condiciones del sistema.

4.2 Registro de centros de acopio.

El sistema debe permitir a un administrador registrar los centros de acopio.

- b) Registro de los datos generales de un centro de acopio.
- c) Investigue cuales son los datos generales de un centro de acopio tales como nombre, dirección, contacto, encargado.

4.3 Registro de entidades autorizadas para canje de puntos.

El sistema debe permitir registrar entidades autorizadas para el canje de los puntos.

- a) Registro de los datos generales de las entidades autorizadas para canje de puntos.
- b) Investigue cuales son los datos generales de una entidad tales como nombre, contacto, encargado, descripción del negocio o producto que ofrecen.

4.4 Registro de puntos.

El sistema debe permitir registrar el reciclaje por las diferentes categorías (plástico, aluminio, etc).

- c) El sistema debe mostrar los valores y el usuario los registra para su almacenamiento en el sistema.

4.5 Parametrización de puntos

El sistema debe registrar y cambiar los parámetros de puntos a partir de la cantidad correcta de reciclaje.

Por ejemplo, el kilo de papel equivale a 10 puntos.

- a) Los puntos son equivalentes a colones.
- b) El sistema debe estar en capacidad de funcionar en otro país con diferentes monedas.
- c) Debe parametrizar el equivalente en la moneda a partir de los kilos de reciclaje de cada tipo (papel, cartón, vidrio, latas, plástico, entre otros. Investigue todos los tipos que existen.

4.6 Productos y canjeo en los comercios autorizados

El sistema debe permitir canjear los puntos por productos.

- a) Los productos deben desplegarse con foto y descripción en cada comercio para que los usuarios los vean y si gustan los seleccionan para el canje.
- b) Al usuario se le permite escoger la cantidad del producto deseado.
- c) El sistema debe validar que el acumulado de puntos alcance para los productos que el usuario desea canjear.

4.7 Jobs

El sistema debe guardar en una bitácora el usuario que generó más reciclaje durante el mes. Debe indicar en cuál centro de acopio lo realizó.

Este llenado de datos se debe hacer por medio de un job que se ejecute el primer día del mes a las 12 a.m.

4.8 Auditoría

El sistema debe manejar campos de auditoría para los registros de datos en la tablas de la base de datos y para la última actualización de un registro. Al menos debe guardar fecha de creación, usuario de creación, fecha de última modificación, usuario que modificó de último cada 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 a la bitácora filtrable por rango de fecha, hora, tipo de cambio, objeto y usuario.

4.9 Control

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

4.10 Módulo de Consultas

Este módulo le permitirá contar con información precisa e inmediata.

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

Las consultas mínimas que debe ofrecer el sistema son:

- a) Listado de comercios afiliados con sus datos y sus direcciones.

La consulta debe incluir un total de comercios al final. Filtros mínimos: nombre del comercio, encargado, tipo de comercio.

- b) Listado de usuarios del sistema. Filtros mínimos: nombre, cédula, provincia, cantón, distrito.
- c) Listado de los usuarios cuyas claves no han sido modificadas en los últimos días o meses. El reporte debe incluir un total de personas al final. Filtros mínimos: nombres, apellidos, cédula, usuario, rango de fechas.
- d) Listado de productos con mayor solicitud de canje y sus respectivos comercios.
- e) Listado de usuarios con total de puntos, cantidad de puntos acumulados, puntos canjeados y diferencia. El reporte debe incluir un total de cada uno de los puntos. Filtros mínimos: rango de fechas.

4.11 Módulo de Consultas para comercios afiliados

Las consultas mínimas que deben mostrarse al usuario son:

- a) Listado de productos que ofrecen.
- b) Top 5 de productos más canjeados.
- c) Total de puntos canjeados en sus productos.

4.12 Módulo de Consultas para centros de acopio

Las consultas mínimas que deben mostrarse a los centros de acopio son:

- a) Listado de materiales reciclados subdividido por tipo de material y cantidad recibida agrupada por mes y año. Filtros mínimos: mes y año.
- b) Top 5 de usuarios que más reciclaje les generan.
- c) Listado de puntos obtenidos por usuarios que llevan su reciclaje al centro de acopio. Filtros: cédula, nombre del usuario.

- a) Total de usuarios agrupado por rango de edad. De 0 a 18, de 19 a 30, de 31 a 45, de 46 a 55, de 56 a 65, de 66 a 75, de 76 a 85, mayores a 85.
- b) Total de comercios afiliados agrupados por tipo de comercio.
- c) Total de productos canjeados por mes y por año.
- d) Top 5 de los usuarios con mayores puntajes a nivel general.

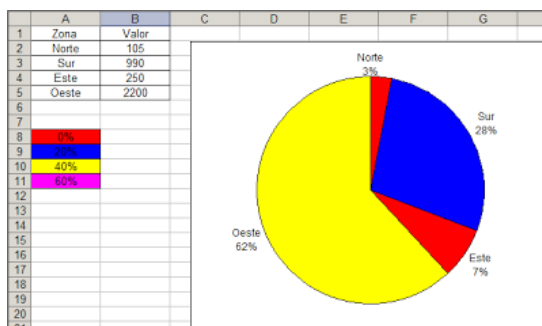
4.13 Filtros

Para todos los filtros: si no se indican datos, el sistema debe retornar todos los registros.

4.14 Módulo de Estadísticas

Este módulo permitirá tener información precisa e inmediata del sistema de reciclaje.

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



Las estadísticas mínimas que debe ofrecer el sistema son:

4.15 Mapa de centros de acopio (opcional)

En este se podrá visualizar de manera gráfica la localización de los centros de acopio a través de la aplicación Google Maps o OpenStreetMap tal y como se muestra en la Figura 2.

Los usuarios tienen acceso a este mapa para que puedan determinar cuál centro de acopio le es más conveniente respecto a ubicación.

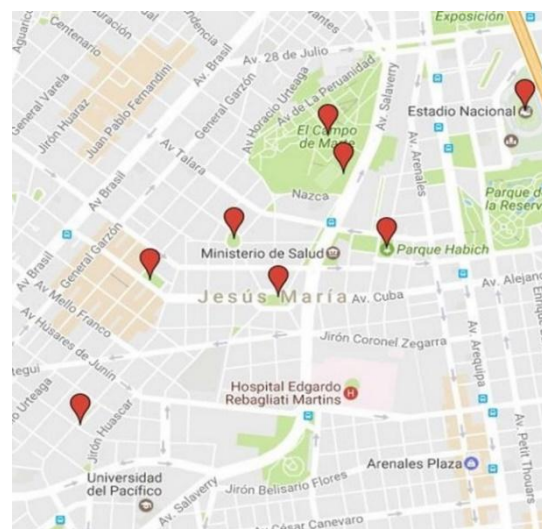


Figura 2 - Ejemplo de marcadores en Google Maps

5 Otras consideraciones

1. El sistema debe presentarse en la defensa con datos reales y bien clasificados con datos de 6 meses.
2. El equipo debe escoger el motor de base de datos que mejor se adapte a este proyecto.
3. El diseño a nivel conceptual y lógico del proyecto debe cumplir con las características fundamentales de bases de datos de integridad, flexibilidad, escalabilidad, seguridad, usabilidad, aspecto visual agradable, fácil modificación y mantenimiento del sistema.
4. La aplicación debe ser gráfica.
5. El sistema debe ser 100% parametrizable.
6. 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.
7. Toda entrada del sistema debe validarse en una salida desde el sistema, es decir, en una consulta, en un reporte o en una estadística. **No es válido ni aceptable que se valide un dato de entrada en la base de datos. La funcionalidad debe ser validada desde el sistema.**
8. No se revisarán sistemas que no logren la integración entre el *front end* y el *back end*. Todas las pruebas deben demostrarse desde el sistema.
9. La aplicación debe guardar las imágenes. Se recomienda en formato Base64. No es indispensable subir las imágenes al Tec Digital pero sí son requisito en la defensa de su proyecto.
10. Usted debe identificar donde aplica crear tablas catálogo o de dominio para que el sistema sea flexible.
11. Se calificará diseño y normalización.
12. Debe utilizar el patrón de diseño *MVC (Model View Controler)*.
13. Todos los valores deben ser parametrizables en el sistema.
14. Se evaluará el diseño y la correcta identificación de los parámetros.
15. Se recomienda trabajar el código desde un programa de control de versiones para demostrar en la defensa la cantidad de commits realizados por cada miembro del equipo. Se recomienda Gitlab.
16. El equipo puede estar conformado por grupos de un número máximo según le indique el profesor.
17. 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.
18. La entrega debe contener los archivos fuente y el ejecutable de la aplicación (si aplica).
19. Cualquier sospecha de copia anulará el trabajo y se procederá con el trámite administrativo.
20. El mejor proyecto tendrá +1 punto extra. Debe estar completo.
21. Si su casa tiene el galardón de Bandera Azul (verificable) obtendrá inmediatamente +1 punto extra.

6 Documentación

La siguiente documentación debe ser entregada:

1. Diagrama de entidad – relación (modelo conceptual). Debe ser en inglés.
2. Manual de usuario con la descripción del propósito y uso del sistema que incluya imágenes significativas de 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*. Los procedimientos, funciones o paquetes importantes deben ir documentados con descripción, el autor principal, fecha de creación.

7 Referencias

Retrived from
<https://banderaazulecologica.org>

Retrived from <https://ecolonescr.com> Plan
Nacional

App Ecoins.

App Reciclemos.