

## Proyecto III – CETracking

Tecnológico de Costa Rica  
Área Académica Ingeniería en Computadores  
CE-1103 Algoritmos y Estructuras de Datos I  
Segundo Semestre 2022  
Valor 15%



### Objetivo General

- Implementar una solución a un problema mediante la utilización de estructuras de datos lineales y grafos.

### Objetivos Específicos

- Diseñar soluciones utilizando el paradigma orientado a objetos.
- Utilizar UML para modelar la solución de un problema.
- Aplicar **patrones de diseño** en la solución de un problema.
- Implementar estructuras de datos lineales y generales.
- Desarrollar las capacidades relacionadas con el trabajo individual y en equipo y el diseño de soluciones.

### Descripción del Problema

CETracking es una aplicación web que permite gestionar la distribución de paquetes entre diferentes centros de almacenamiento y distribución.

Las características principales de la aplicación son las siguientes:

1. **Gestionar mapa.** Para esto se deben permitir las siguientes opciones:
  - a. Generar un nuevo mapa (grafo) con una cantidad aleatoria de centros de distribución/almacenamiento y que a su vez establezca, de forma aleatoria, los centros que tienen comunicación y los respectivos costos de estas aristas.
  - b. Agregar centros de distribución, para lo cual se debe indicar un nombre, establecer con cuáles centros de distribución tiene comunicación y los costos de estas aristas.
  - c. Editar centros de distribución, lo cual implica modificar el nombre, agregar/editar/eliminar aristas que involucren al centro de distribución bajo edición, así como modificar los costos asociados a las aristas.
  - d. Eliminar centros de distribución, para lo cual se deben eliminar toda la información relacionada con el centro de distribución.
2. **Solicitar el envío de un paquete.** La aplicación debe permitir solicitar el envío de un paquete desde un centro de distribución a otro centro de distribución, para lo cual debe establecerlo a través de la aplicación. Para esto se deben realizar los siguientes pasos:
  - a. Seleccionar el punto de inicio y punto de entrega.
  - b. Calcular las diferentes rutas posibles y asignar un costo monetario para cada ruta, considerando el costo en tiempo (segundos) de las aristas que forman parte de la ruta. La lista de opciones de rutas debe mostrarse de forma ordenada. Cada grupo seleccionará el algoritmo de ordenamiento para esta característica.



- c. Asignar un código numérico hexadecimal para cada solicitud de traslado que se realice en el sistema. Este código debe tener obligatorio un valor numérico y otro correspondiente a una letra. Este código se genera de forma aleatorio cuando se hace la solicitud de traslado de un paquete. Se debe validar que no hayan códigos repetidos, por lo que se debe implementar una lista enlazada que permita gestionar esta validación.
- 3. **Visualizar el estado del envío de cada paquete.** Esto involucra:
  - a. Ver en tiempo real el desplazamiento de todos los paquetes que han sido solicitados a través de la aplicación.
- 4. **Dashboard.** Que permita visualizar los siguientes indicadores, con información en tiempo real:
  - a. Total de paquetes que han sido solicitados a través del sistema.
  - b. Total de paquetes entregados.
  - c. Total de paquetes pendientes de entregar.

Este proyecto implica una aplicación web donde se debe permitir la interacción del usuario con las funcionalidades. El cliente es solamente una interfaz donde el usuario interactúa con el sistema, mientras que el servidor es el encargado de la gestión de las estructuras de datos, los algoritmos y el procesamiento de las solicitudes de los usuarios. El cliente solo captura los requerimientos de los usuarios y muestra el resultado de las solicitudes.

#### *CETracking Controller*

Implementado en arduino. Las funcionalidades principales son:

- 1. **Notificar cada paquete entregado en tiempo real.** Esto involucra que en el momento que un paquete llegue a su punto de destino debe:
  - a. Mostrar en un siete segmentos doble el código alfanumérico del envío del paquete.
  - b. Indicar, mediante una alarma (sonido), que un paquete ha llegado. Utilizar un buzzer.
  - c. La notificación mediante el siete segmentos doble y el sonido debe durar 10 segundos. Si otro paquete llega antes de que pasen estos 10 segundos, se debe almacenar en una cola, implementada mediante un arreglo de máximo 10 posiciones.

#### **Requerimientos no funcionales**

- 1. Cabe resaltar que es de manera obligatoria realizar la maqueta del control para interactuar con la aplicación. No se permite presentar el circuito únicamente en la protoboard.
- 2. Bibliotecas para comunicación serial con Arduino: jSerialComm o RXTX.

#### **Documentación requerida**

- 1. Internamente, el código se debe documentar utilizando Javadoc. Se debe generar el HTML.
- 2. La documentación externa se hará en un documento que incluya lo siguiente (deberá entregarse un PDF):
  - a. Breve descripción del problema.
  - b. Diagrama de clases.
  - c. Descripción de las estructuras de datos desarrolladas.
  - d. Descripción de los algoritmos desarrollados.
  - e. Problemas encontrados en forma de bugs de *github*: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.



3. **Planificación y administración del proyecto:** se utilizará Azure DevOps para la administración del proyecto. Debe incluir:
  - a. Lista de features e historias de usuario identificados de la especificación.
  - b. Plan de iteraciones que agrupen cada bloque de historias de usuario de forma que se vea un desarrollo incremental
  - c. Descomposición de cada user story en tareas.
  - d. Asignación de tareas entre los integrantes del grupo.

**Aspectos operativos y evaluación:**

1. **Fecha de entrega: De acuerdo con el cronograma del curso y lo establecido en el TEC Digital**
2. El proyecto tiene un valor de 15% de la nota del curso.
3. El trabajo es **en grupos de cuatro estudiantes.**
4. Deben entregar en el TEC Digital un documento con el link del repositorio de GitHub, Azure DevOps y el PDF de la documentación. Para ambas herramientas deben dar acceso al correo del profesor.
5. Es obligatorio utilizar un Git y GitHub para el control de versiones del código fuente y evidenciar el uso de Commits frecuentes.
6. Es obligatorio integrar toda la solución.
7. El código tendrá un valor total de 70%, la documentación externa 20% y la defensa un 10%.
8. De las notas mencionadas en el punto anterior se calculará la Nota Final del Proyecto.
9. Se evaluará que la documentación sea coherente, acorde a la dificultad/tamaño del proyecto y el trabajo realizado. Se recomienda que realicen la documentación conforme se implementa el código.
10. La nota de la documentación externa es proporcional a la completitud del proyecto.
11. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
12. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la primera cita de revisión oficial.
13. Aún cuando el código y la documentación externa tienen sus notas por separado, se aplican las siguientes restricciones
  - a. Si no se entrega documentación externa, automáticamente se obtiene una nota de cero en la nota final del proyecto.
  - b. Si no se utiliza un manejador de código se obtiene una nota de cero en la nota final del proyecto.
  - c. Si la documentación externa no se entrega en la fecha indicada se obtiene una nota de cero en la nota final del proyecto.
  - d. Si el código no compila se obtendrá una nota de cero en la nota final del proyecto, por lo cual se recomienda realizar la defensa con un código funcional.
  - e. El código debe ser desarrollado en Java (Windows), en caso contrario se obtendrá una nota de cero en la nota final del proyecto.
  - f. Si alguna persona integrante del proyecto no se presenta a la revisión se asignará una nota de cero en la nota final del proyecto.
14. La revisión de la documentación podrá ser realizada por parte del profesora antes, durante o después de la revisión del proyecto.



15. Cada estudiante tendrá como máximo 30 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.
16. 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.
17. Cada estudiante es responsable de llevar los equipos requeridos para la revisión, si no cuentan con estos deberán avisar al menos 2 días antes de la revisión al profesor para coordinar el préstamo de estos.
18. Durante la revisión únicamente podrán participar el estudiante, asistentes, otros profesores y el coordinador del área.



## **ANEXO DEL PROYECTO (2.5% extra del Proyecto)**

### **Objetivo General**

- Elaborar un documento que evidencie el establecimiento de los lineamientos adecuados para el trabajo en equipo, así como la identificación de las responsabilidades asociadas a cada rol.

### **Objetivos Específicos**

- Determinar los requerimientos de ingeniería para un problema complejo considerando partes involucradas, estado del arte, estándares, normas, entre otros.
- Establecer lineamientos adecuados para el trabajo en equipo (metas, roles, reglas, cronogramas, bitácoras, otros).
- Identificar roles y sus respectivas responsabilidades como miembro de un equipo de trabajo.

### **Atributos de Acreditación**

- Diseño (Inicial).
- Trabajo individual y en equipo (Inicial).

### **Descripción del Entregable**

Cada grupo debe elaborar un documento que tenga la siguiente estructura:

1. Portada.
2. Tabla de contenidos.
3. Introducción.
4. Diseño (Determinar los requerimientos de ingeniería para un problema complejo considerando partes involucradas, estado del arte, estándares, normas, entre otros):
  - a. Listado de historias de usuario.
  - b. Estado del arte: Describir los conceptos teóricos asociados al proyecto con la respectiva referencia bibliográfica para cada concepto.
5. Trabajo en equipo (evidencia de la participación activa en el equipo de trabajo de acuerdo con el rol asignado y la ejecución adecuada de los lineamientos para el trabajo en equipo: metas, roles, reglas, cronogramas, bitácoras, entre otros):
  - a. Metas del proyecto: Describir los entregables asociados al proyecto.
  - b. Roles: Describir los roles y responsabilidades utilizados en el desarrollo del proyecto.
  - c. Reglas: Describir las reglas principales que han sido definidas para el trabajo en equipo.
  - d. Cronograma: Plan de proyecto con las actividades planeadas, responsables de cada actividad y fechas de entregas estimadas.
  - e. Bitácora: Evidencia la participación activa en el trabajo en equipo y la ejecución adecuada de los lineamientos para el trabajo en equipo, de cada uno de los integrantes del equipo. Deben describir las actividades realizadas como reuniones con el compañero de trabajo, investigaciones, consultas, entre otras. Se debe describir todo por más insignificante que sea. Esto demostrará el trabajo de cada uno de los miembros del equipo según el rol asignado.

**Aspectos operativos y evaluación**

1. El valor extra del anexo sólo será válido si se hace entrega de la documentación externa, de la aplicación web y lo correspondiente al arduino.
2. El documento debe ser enviado en formato PDF.