

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Introducción a la Programación y Computación 2

Ing. Claudia Liceth Rojas Morales
Ing. Marlon Antonio Pérez Türk
Ing. José Manuel Ruiz Juárez
Ing. Edwin Estuardo Zapeta Gómez
Ing. Byron Rodolfo Zepeda Arevalo

Tutores de curso:

Josué Alfredo González
Marvin Daniel Rodríguez
Daniel Arturo Alfaro
Sergio Felipe Zapeta
Erwin Fernando Vásquez



PROYECTO No. 2

OBJETIVO GENERAL

Modelar, documentar e implementar una solución al problema que se plantea utilizando las herramientas de desarrollo presentadas en clase y laboratorio.

OBJETIVOS ESPECÍFICOS

- Implementar una solución utilizando el lenguaje de programación Python.
- Utilizar estructuras de programación secuenciales, cíclicas y condicionales.
- Generar reportes con la herramienta Graphviz.
- Manipular archivos XML.
- Utilizar los conceptos de TDA y aplicarlos a memoria dinámica.
- Utilizar estructuras de programación propias.
- Utilizar el paradigma de programación orientada a objetos.

DESCRIPCIÓN DEL PROBLEMA

El ministerio de la Defensa de Guatemala ha creado un acuerdo de colaboración con la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala. El objetivo de este acuerdo es crear una nueva tecnología que permita el envío de mensajes encriptados de tal forma que no puedan ser interceptados y descifrados por personal o instituciones no autorizadas.

Para lograr este fin, se ha creado un sistema con 2 componentes. Un componente emisor del mensaje y un componente receptor del mensaje.

La Facultad de Ingeniería propone un diseño que funcionará con “n” drones, los cuales podrán subir una cantidad de metros y emitir una luz led de alta emisión de tal forma que dependiendo del dron que emita la luz y la altura a la que la emita, representará una letra del alfabeto, de esta forma, el componente receptor podrá medir alturas y determinar qué dron emitió el haz de luz, decodificar cada letra y finalmente obtener el mensaje que se desea transmitir.

Cada día, el ministerio de defensa creará una tabla como la siguiente:

8		X	Y	Z
7	T	U	V	W
6	M	Q	R	S
5	N	Ñ	O	P
4	K	L	L	E
3	H	I	A	J
2	E	F	G	A
1	A	B	C	D
Altura (mts)	Dron01	Dron02	Dron03	Dron04

Figura 1 – Sistema de drones generado por el ejército de Guatemala
(**OJO:** El Dron01 a 8 metros representa un espacio en blanco)

Cada dron es capaz de realizar las siguientes acciones:

1. Subir 1 metro
2. Bajar 1 metro

3. Esperar
4. Emitir luz de alta energía

El sistema de drones debe respetar las siguientes reglas:

- a. Solamente un dron puede “Emitir luz de alta energía” en un tiempo “t” dado.
- b. Un dron puede emitir la misma letra a distintas alturas, así será más difícil hackear el sistema de encriptación producido por los drones.
- c. Un dron demora 1 segundo en subir o bajar un metro y demora 1 segundo en encender y apagar la luz de alta energía.

Para crear un mensaje a transmitir, simplemente se transmite al sistema de drones una secuencia de instrucciones que determina el orden en que deben encender sus luces cada uno de los drones y la altura a la que lo deben hacer. Así, por ejemplo, si el mensaje es “Hello World” se podría enviar la secuencia de instrucciones que se muestra en la tabla 1 al sistema de drones:

Dron01,3	-	Dron01 a 3 metros (representa la H)
Dron04,4	-	Dron04 a 4 metros (representa la E)
Dron03,4	-	Dron03 a 4 metros (representa la L)
Dron02,4	-	Dron02 a 4 metros (representa la L)
Dron03,5	-	Dron03 a 5 metros (representa la O)
Dron01,8	-	Dron01 a 8 metros (representa el espacio en blanco)
Dron04,7	-	Dron04 a 7 metros (representa la W)
Dron03,5	-	Dron03 a 5 metros (representa la O)
Dron03,6	-	Dron03 a 6 metros (representa la R)
Dron03,4	-	Dron03 a 4 metros (representa la L)
Dron04,1	-	Dron04 a 1 metro (representa la D)

Tabla 1 – Ejemplo de instrucciones que recibe un sistema de 4 drones para emitir el mensaje “Hello World”

Debido a que un sistema de drones puede tener una letra en varias alturas y/o en diferentes drones, entonces, un mismo mensaje podría ser enviado con distintas instrucciones.

El sistema receptor, será capaz de detectar las alturas y los nombres de los drones, de esta forma, podrá buscar en el sistema de drones proporcionado por el Ministerio de Defensa, la letra correspondiente y reconstruir el mensaje.

Usted ha sido contratado por la Facultad de Ingeniería de USAC para desarrollar un Software capaz de manejar el sistema de drones que genera el mensaje, y en base a las instrucciones recibidas, generar el mensaje en el menor tiempo posible¹.

El sistema deberá ser capaz de manejar hasta 200 drones y alturas de 1 a 100 metros.

¹ OJO: Debe considerar que debe respetar las reglas definidas para el sistema de drones.

Ejemplo de funcionalidad para 3 drones con un límite de 7 metros de altura

Definición del sistema de drones

Altura (mts)	DronX	DronY	DronZ
1	A	2	B
2	I	C	C
3	D	P	E
4	F	G	H
5	I	L	O
6	J	M	P
7	K	N	Q

Mensaje que se desea enviar al sistema de drones

Instrucción
DronX,2
DronY,3
DronZ,2
DronY,1

Instrucciones detalladas que se deben enviar al sistema de drones para que emita el mensaje encriptado (optimizando el tiempo para generar el mensaje)

Tiempo (seg)	DronX	DronY	DronZ
1	Subir	Subir	Subir
2	Subir	Subir	Subir
3	Emitir luz	Subir	Esperar
4	Esperar	Emitir luz	Esperar
5	Esperar	bajar	Emitir luz
6	Esperar	bajar	Esperar
7	Esperar	Emitir luz	Esperar

En este ejemplo, el tiempo óptimo para emitir el mensaje haciendo que el sistema de drones encienda sus luces en las alturas determinadas por el set de instrucciones es de 7 segundos.

El mensaje enviado es el siguiente: IPC2

ENTRADA

Se podrá utilizar un archivo en formato XML para crear una configuración inicial que facilite la evaluación de la funcionalidad del software desarrollado.

entrada.xml

```
<?xml version="1.0"?>
<config>
  <listaDrones>
    <dron> [valorAlfanumerico] </dron>
    ...
  </listaDrones>
  <listaSistemasDrones>
    <sistemaDrones nombre="NombreSistemaDrones">
      <alturaMaxima>[valorNumerico]</alturaMaxima>
      <cantidadDrones>[valorNumerico]</cantidadDrones>
      <contenido>
        <dron> [valorAlfanumerico] </dron>
        <alturas>
          <altura valor="valorAltura"> [valorAlfanumerico] </altura>
          ...
        </alturas>
      </contenido>
      ...
    </sistemaDrones>
    ...
  </listaSistemasDrones>
  <listaMensajes>
    <Mensaje nombre="NombreMensaje">
      <sistemaDrones>[valorAlfanumerico]</sistemaDrones>
      <instrucciones>
        <instruccion dron="NombreDron">[valorNumericoAltura]</instruccion>
        ...
      </instrucciones>
    </Mensaje>
    ...
  </listaMensajes>
</config>
```

Tomar en cuenta que este archivo es incremental, es decir, se pueden ingresar varios archivos de entrada, de tal manera que se pueden crear nuevos drones, nuevos sistemas de drones y nuevos mensajes.

SALIDA

Se debe generar un archivo en formato XML que muestre el detalle de instrucciones para enviar los mensajes configurados en los distintos sistemas de drones y el tiempo óptimo necesario para enviar dichos mensajes.

Salida.xml

```
<?xml version="1.0"?>
<respuesta>
  <listaMensajes>
    <mensaje nombre="nombreMensaje">
      <sistemaDrones>[valorAlfanumerico]</sistemaDrones>
      <tiempoOptimo>[valorNumerico]</tiempoOptimo>
      <mensajeRecibido>[valorAlfanumerico]</mensajeRecibido>
      <instrucciones>
        <tiempo valor="valorNumerico">
          <acciones>
            <dron nombre="valorAlfanumerico"> [valorAccionDron] </dron>
            ...
          </acciones>
        </tiempo>
        ...
      </instrucciones>
    </mensaje>
    ...
  </listaMensajes>
</respuesta>
```

INTERFAZ DE USUARIO

Se debe crear una interfaz de usuario fácil de utilizar e intuitiva que permita al usuario realizar las siguientes funciones:

- a. Inicialización – Para que el sistema pueda inicializarse sin ninguna información previa.
- b. Cargar un archivo XML de entrada
- c. Generar un archivo XML de salida
- d. Gestión de drones
 - a. Ver listado de drones ordenado alfabéticamente
 - b. Agregar un nuevo dron (debe ser un nombre único en el software)
- e. Gestión de sistemas de drones
 - a. Ver gráficamente listado de sistemas de drones (utilizando Graphiz)
- f. Gestión de Mensajes
 - a. Ver listado de mensajes y sus instrucciones (ordenado alfabéticamente por nombre del mensaje)
 - b. Ver instrucciones para enviar un mensaje
 - i. Seleccionar un mensaje
 - ii. Mostrar el nombre sistema de drones a utilizar, el mensaje que se enviará y el tiempo óptimo para que el sistema de drones muestre el mensaje
 - iii. Ver gráficamente (utilizando Graphiz) el listado de instrucciones que se debe enviar al sistema de drones para que genere el mensaje en un tiempo óptimo.
- g. Ayuda – Mostrar la información del estudiante y un link hacia la documentación del proyecto.

CONSIDERACIONES

Se deberá realizar la implementación utilizando programación orientada a objetos, algoritmos desarrollados por el estudiante e implementación de estructuras a través de Tipos de Dato Abstracto (TDAs) propios del estudiante, es decir, no deben utilizar estructuras de datos propias de Python (list, dict, tuple, set). El estudiante deberá abstraer la información y definir qué estructuras implementar para construir la solución.

Debe utilizarse versionamiento para el desarrollo del proyecto. Se utilizará la plataforma **GitHub** en la cual se debe crear un repositorio en el que se gestionará el proyecto. Se deben realizar **4 releases** o versiones del proyecto (Se sugiere desarrollar un Release por semana). Se deberá agregar a sus respectivos auxiliares como colaboradores del repositorio. El último release será el release final y se deberá de realizar antes de entregar el proyecto en la fecha estipulada.

DOCUMENTACIÓN

Para que el proyecto sea calificado, el estudiante deberá entregar la documentación utilizando el formato de ensayo definido para el curso. En el caso del proyecto, el ensayo puede tener un mínimo de 4 y un máximo de 7 páginas de contenido, este máximo no incluye los apéndices o anexos donde se pueden mostrar modelos y diseños utilizados para construir la solución. Es obligatorio incluir el diagrama de clases del diseño del software desarrollado y se recomienda incluir el modelo conceptual y los diagramas de actividades que modelan la solución de software presentada por el estudiante.

RESTRICCIONES

- Solo se permitirá la utilización de los IDEs discutidos en el laboratorio.
- Uso de TDA implementados por el estudiante obligatorio; el uso de estructuras de Python (list, dict, tuple, set) resultará en penalización del 100% de la nota.
- Uso obligatorio de programación orientada a objetos (POO) desarrollada por completo por el estudiante. De no cumplir con la restricción, no se tendrá derecho a calificación.
- El nombre del repositorio debe de ser **IPC2_Proyecto2_#Carnet**.
- El estudiante debe entregar la documentación solicitada para poder optar a la calificación.
- Los archivos de entrada no podrán modificarse.
- Los archivos de salida deben llevar la estructura mostrada en el enunciado obligatoriamente.
- Deben existir 4 releases mínimo, podría ser uno por cada semana, de esta manera se corrobora el avance continuo del proyecto.
- Se calificará de los cambios realizados en el último release hasta la fecha de entrega. Los cambios realizados después de ese release no se tomarán en cuenta.
- Cualquier caso de copia parcial o total tendrá una nota de 0 y será reportada a Escuela.
- Para dudas concernientes al proyecto se utilizarán los foros en UEDI de manera que todos los estudiantes puedan ver las preguntas y las posteriores respuestas.
- **NO HABRÁ PRÓRROGA.**

ENTREGA

- La entrega será el domingo **01 de octubre**, a más tardar a las 11:59 pm.
- La entrega será por medio de la UEDI.
- La documentación debe estar subida en el repositorio en una carpeta separada.
- Para entregar el proyecto en UEDI se deberá subir un archivo de texto con el link del repositorio.