



PROGRAMACIÓN Y MODELADO EN INGENIERÍA BIOQUÍMICA

INTEGRANTES DEL EQUIPO (EN ORDEN ALFABÉTICO POR APELLIDO, MÁXIMO CINCO)	NÚMERO DE CONTROL

Intención didáctica

Obtener una perspectiva de la importancia y aplicación de la programación y el modelado matemático en la ingeniería bioquímica.

Indicaciones

1. Mediante una investigación bibliográfica, preparar un ensayo presentando su perspectiva respecto al desarrollo histórico de las computadoras y su importancia en ingeniería.
2. Investigar al menos tres ejemplos de simulaciones por computadora que estén relacionadas con la ingeniería bioquímica.
3. (OPCIONAL) Diseñar una encuesta y aplicarla a un mínimo de 25 compañeros de la carrera, para obtener al menos la siguiente información: qué lenguajes de programación han usado, y en qué clases los han aplicado.
4. (OPCIONAL) Entrevistar a algún profesor de la carrera, para obtener su opinión respecto al uso de la computadora como herramienta de simulación en ingeniería bioquímica.

Sugerencias para el éxito de la actividad

- ★ El diseño de la encuesta y de la entrevista es libre; es una oportunidad para evidenciar su creatividad ©.
- ★ Los resultados de la encuesta deben presentarse en forma agregada, auxiliándose de gráficas.

Evidencias Entregables

El reporte de esta actividad lleva esta hoja de instrucciones como portada y su extensión máxima es de 10 páginas, distribuidas aproximadamente de la siguiente forma: ensayo (4-5 páginas, incluir referencias bibliográficas), ejemplos de simulación en ingeniería bioquímica (1 página), el formulario de la encuesta o la lista de preguntas (una página), los resultados de la encuesta (1-2 páginas), entrevista (1 página) y sus conclusiones individuales de la actividad (1 página).

Sólo se entrega un ejemplar del reporte por equipo. Una vez revisado, cada integrante del equipo deberá tener una copia para su portafolio final.



ALGORITMOS Y DIAGRAMAS DE FLUJO

INTEGRANTES DEL EQUIPO (POR APELLIDO, EN ORDEN ALFABÉTICO, MÁXIMO TRES)	NÚMERO DE CONTROL

Intención didáctica

Familiarizarse con el concepto de algoritmo, la importancia de que esté correctamente especificado, y la simbología usada en los diagramas de flujo de programación.

Indicaciones

1. Seleccionar un lugar en la zona sur del ITD. **Este lugar es su objetivo.** Es importante que no den a conocer el lugar que seleccionaron a los otros equipos.
ANOTAR AQUÍ SU LUGAR OBJETIVO: _____
2. Iniciando desde el lugar indicado por el profesor, realizar el recorrido hasta su objetivo, anotando en la hoja anexa una lista numerada de todas las acciones necesarias para llegar a ese lugar.
3. Una vez de regreso en el punto de partida, recibirán la hoja de instrucciones de otro equipo, y deberán realizar el recorrido siguiendo al pie de la letra las instrucciones. Anotar el lugar al cual llegaron, y si tuvieron alguna dificultad para llegar.
4. Regresar de nuevo al punto de partida para comparará el lugar al que llegaron con el lugar objetivo, para determinar si el algoritmo para llegar ahí estuvo correctamente especificado. Establecer una conclusión por equipo respecto a la importancia de que los algoritmos que estén adecuadamente especificados.
5. Realizar una investigación bibliográfica (máximo tres páginas) acerca de qué es un algoritmo, qué es un pseudo-código, para qué sirve un diagrama de flujo y qué simbología emplea. Incluir al menos un ejemplo de cada concepto.

Evidencias Entregables

El reporte de esta actividad lleva esta hoja de instrucciones como portada, seguida de la hoja de trabajo con su algoritmo y los comentarios del equipo que probó su algoritmo. Después, incluir su investigación bibliográfica y las conclusiones del equipo sobre la actividad.

Sólo se entrega un ejemplar del reporte por equipo. Una vez revisado, cada integrante del equipo deberá tener una copia para su portafolio final.



ALGORITMOS Y DIAGRAMAS DE FLUJO – HOJA DE TRABAJO

EQUIPO QUE ELABORÓ EL ALGORITMO (POR APELLIDO, EN ORDEN ALFABÉTICO, MÁXIMO TRES)	NÚMERO DE CONTROL

ALGORITMO PARA LLEGAR A UN LUGAR EN LA ZONA SUR DEL ITD (NO ANOTAR A CUÁL LUGAR):

(si necesitan espacio adicional, usar el reverso de la hoja)

EQUIPO QUE PROBÓ EL ALGORITMO (EN ORDEN ALFABÉTICO POR APELLIDO, MÁXIMO TRES)	NÚMERO DE CONTROL

LUGAR AL QUE LLEGARON: _____

DIFICULTADES O COMENTARIOS:



COMPARACIÓN DE LENGUAJES DE PROGRAMACIÓN

INTEGRANTES DEL EQUIPO (POR APELLIDO, EN ORDEN ALFABÉTICO, MÁXIMO TRES)	NÚMERO DE CONTROL

Intención didáctica

Apreciar las diferencias y similitudes que hay entre varios lenguajes de programación.

Antecedentes

En clase se han visto algunos aspectos de programación en el lenguaje Scilab. Prácticamente todos los lenguajes de programación implementan estas características, pero frecuentemente de una forma diferente.

Indicaciones

1. Investigar los operadores aritméticos, de comparación y booleanos en un mínimo de tres lenguajes de programación diferentes (excepto Scilab). Ejemplos de lenguajes que pueden consultar son C, C++, Java, Visual Basic, Fortran, Python, etcétera. Elaborar una tabla en la que presenten los operadores que tengan la misma función, cada lenguaje en una columna, donde la primera columna debe ser los operadores de Scilab. Por ejemplo:

Operador	Scilab	VB	C++	Fortran
“y” lógico	&	And	&&	.AND.

La tabla debe incluir, como mínimo, todos los operadores que se vieron en clase para Scilab.

2. Para cada uno de los lenguajes que seleccionaron en el paso anterior, buscar un programa de ejemplo donde aparezca el mayor número posible de las estructuras estudiadas en esta unidad: **decisión**, **selección**, **ciclos** y **funciones** (en Scilab serían, respectivamente, **if**, **select**, **for** y **while**, y **function**). Señalar claramente cada una de estas estructuras en el ejemplo.

Evidencias Entregables

El reporte de esta actividad lleva esta hoja de instrucciones como portada. A continuación debe llevar la tabla de los operadores (aritméticos, de comparación y lógicos), y los ejemplos de programas en cada uno de los lenguajes, señalando las estructuras de control y funciones. Finalmente incluir comentarios y conclusiones (del equipo) respecto a las similitudes y diferencias entre los lenguajes de programación investigados.

Sólo se entrega un ejemplar del reporte por equipo. Una vez revisado, cada integrante del equipo deberá tener una copia para su portafolio final.