TRABAJO TEÓRICO PRÁCTICO: INNOVATIVE AND ROUTINE DESIGN.

INGENIERÍA DE SISTEMAS, TUPAR, TUDAI.

➢ Actividades de pre-lectura

1. **AISLE** y **ESCRIBA** las palabras transparentes que encuentre en el texto.
2. **RELEA** la lista y **REDACTE** una hipótesis de lo que tratará el texto.
3. **RELACIONE** las palabras del cuadro con sus significados en castellano.

A. Several	varios	***1. ambos***
B. latter	ultimos	***2.últimos***
C.carriers	transmisores	***3.dar***
D.both	ambos	***4.elemento de primera necesidad***
E. fail	fracasar	***5. transmisores***
F. staple	elemento de primera necesidad	***6.varios***
G. provide	dar	***7.fracasar***

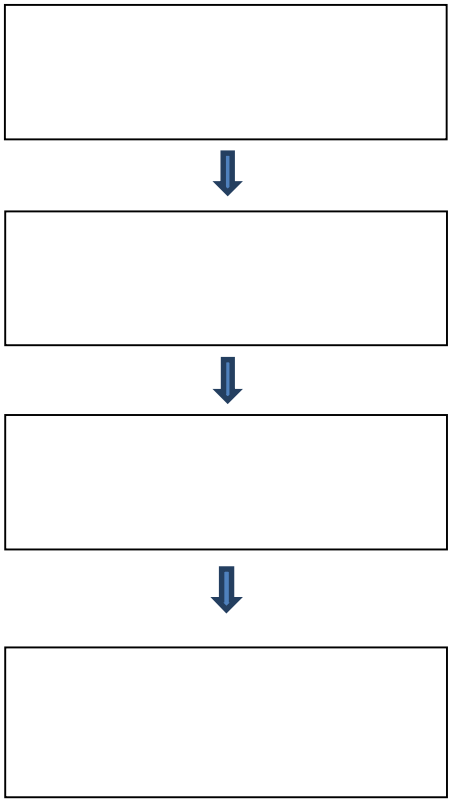
➢ Actividades de lectura

4. ESCRIBA verdadero (V) o falso (F).

a) FALSO La mayor parte del trabajo de diseño está dedicada a diseños de rutina.
b) VERD La ingeniería del software capta, organiza y comparte el conocimiento de diseño para hacer el diseño de rutina más simple
c) ..... El diseño de software termina casi siempre siendo innovador gracias a la recopilación de información acerca de los diseños de rutina.
d) ..... La programación se ha visto beneficiada durante mucho tiempo por la codificación y el almacenamiento de diseños en bibliotecas de subrutinas.

5. EXPLIQUE con sus palabras qué entiende por la expresión idiomática al final del primer párrafo ***"the latter are the bread and butter of engineering***".

6- COMPLETE los casilleros con una frase que sintetice la idea de cada párrafo y le permita obtener un resumen de lo leído.



➢ Gramática

- ***Oraciones condicionales***

7. OBSERVE las dos oraciones que se consignan ***en color verde en el texto***. Ambas poseen o expresan una condición[1]. **INDIQUE** en qué lugar de cada oración se expresa esa idea.

¿De cuántas partes constan? **IDENTIFÍQUELAS**.

¿Qué forma verbal puede reconocer en cada una de sus partes?

¿Cómo traduciría o interpretaría cada ejemplo?

- ***Recursos Deícticos***

8. ¿A qué palabras del texto hacen referencia los vocablos que en el artículo aparecen marcados con un círculo?

Problema 5:
Los diseños originales son ucho menos necesario que los diseños de rutina, por lo que este último es el pan y la manteca de la ingenieria.
HICE UNA TRADUCCION TEXTUAL. NUNCA HABIA ESCUCHADO ESE DICHO, POR LO QUE DESCONOZCO QUE SE QUISO DECIR.

Problema 8:
ONE: hace referencia a DISTINCTIONS. Recusrso CATAFORICO
THE LATTER: hace referencia a ROUTINE DESINGS. Recurso ANAFORICO
THEY: hace referencia a current notations for software designs. Recurso ANAFORICO
SO: hace refencia a
IT: hace referencia a this knowledge. Recurso ANAFORICO

[1] Para realizar esta actividad, recuerde ver el video correspondiente.

Routine and InnovativeDesign

Engineering design tasks are of several kinds; one of the most significant distinctions separates routine from innovative design. Routine design involves solving familiar problems, reusing large portions of prior solutions. Innovative design, on the other hand, involves finding novel solutions to unfamiliar problems. Original designs are much more rarely needed than routine designs, so the latter is the bread and butter of engineering.

Most engineering disciplines capture, organize, and share design knowledge in order to make routine design simpler. Handbooks and manuals are often the carriers of this organized information [Marks 87, Perry 84]. But current notations for software designs are not adequate for the task of both recording and communicating designs, so they fail to provide a suitable representation for such handbooks. Software in most application domains is treated more often as original than routine— certainly more so than would be necessary if we captured and organized what we already know. One path to increased productivity is identifying applications that could be routine and developing appropriate support. The current focus on reuse emphasizes capturing and organizing existing knowledge of a particular kind: knowledge expressed in the form of code. Indeed, subroutine libraries—especially of system calls and general-purpose mathematical routines—have been a staple of programming for decades. But this knowledge cannot be useful if programmers do not know about it or are not encouraged to use it. Further, library components require more care in design, implementation and documentation than similar components that are simply embedded in systems. Practitioners recognize the need for mechanisms to share experience with good designs. This cry from the wilderness appeared on a Software Engineering news groups:

"In Chem E, when I needed to design a heat exchanger, I used a set of references that told me what the constants were... and the standard design equations...

"In general, unless I, or someone else in my engineering group, has read or remembers and makes known a solution to a past problem, I'm doomed to recreate the solution. ... I guess ... the critical difference is the ability to put together little pieces of the problem that are relatively well known, without having to generate a custom solution for every application...

"I want to make it clear that I am aware of algorithm and code libraries, but they are incomplete solutions to what I am describing. (There is no Perry's Handbook for Software Engineering.)"

This former chemical engineer is complaining that software lacks the institutionalized mechanisms of a mature engineering discipline for recording and disseminating demonstrably good designs and ways to choose among design alternatives. Perry's handbook is the standard design handbook for chemical engineering; it is about 4 inches thick x 8-1/2" x 11", printed in tiny type on tissue paper [Perry 84].

Fuente:

Shaw, M. (1990). *Prospects for an Engineering Discipline of Software*. pp.2-3.

➢ ACTIVIDADES DE ESCRITURA

La elaboración de resúmenes constituye una práctica de estudio usual en los ámbitos académicos y; al momento de elaborarlos, el lector necesita reducir la información del texto leído. Reducir o suprimir información requiere mantener sólo las proposiciones que son textualmente pertinentes y omitir las proposiciones, frases, y/ o apartados que se repiten o que incorporan o introducen detalles, ejemplificaciones, etc.

9- UTILICE el texto en blanco de las páginas 6 y 7. **TACHE/ BORRE** o **UTILICE** paréntesis para omitir toda aquella información que resulte innecesaria. **EXPLIQUE** porqué omitió lo consignado.

Routine and Innovative Design

Engineering design tasks are of several kinds; one of the most significant distinctions separates routine from innovative design. Routine design involves solving familiar problems, reusing large portions of prior solutions. Innovative design, on the other hand, involves finding novel solutions to unfamiliar problems. Original designs are much more rarely needed than routine designs, so the latter is the bread and butter of engineering.

Most engineering disciplines capture, organize, and share design knowledge in order to make routine design simpler. Handbooks and manuals are often the carriers of this organized information [Marks 87, Perry 84]. But current notations for software designs are not adequate for the task of both recording and communicating designs, so they fail to provide a suitable representation for such handbooks. Software in most application domains is treated more often as original than routine— certainly more so than would be necessary if we captured and organized what we already know. One path to increased productivity is identifying applications that could be routine and developing appropriate support. The current focus on reuse emphasizes capturing and organizing existing knowledge of a particular kind: knowledge expressed in the form of code. Indeed, subroutine libraries—especially of system calls and general-purpose mathematical routines—have been a staple of programming for decades. But this knowledge cannot be useful if programmers do not know about it or are not encouraged to use it. Further, library components require more care in design, implementation and documentation than similar components that are simply embedded in systems. Practitioners recognize the need for mechanisms to share experience with good designs. This cry from the wilderness appeared on a Software Engineering news groups:

"In Chem E, when I needed to design a heat exchanger, I used a set of references that told me what the constants were ... and the standard design equations ...

"In general, unless I, or someone else in my engineering group, has read or remembers and makes known a solution to a past problem, I'm doomed to recreate the solution. ... I guess ... the critical difference is the ability to put together little pieces of the problem that are relatively well known, without having to generate a custom solution for every application...

"I want to make it clear that I am aware of algorithm and code libraries, but they are incomplete solutions to what I am describing. (There is no Perry's Handbook for Software Engineering.)"

This former chemical engineer is complaining that software lacks the institutionalized mechanisms of a mature engineering discipline for recording and disseminating demonstrably good designs and ways to choose among design alternatives. Perry's handbook is the standard design handbook for chemical engineering; it is about 4 inches thick x 8-1/2" x 11", printed in tiny type on tissue paper [Perry 84].

11-UTILICE el comienzo sugerido y las preguntas orientativas para redactar un párrafo. Recuerde presentar la información de manera académica, tal como hemos visto en la clase anterior.

En ingeniería, existen dos tipos de tareas relativas al diseño: las de rutina y las innovadoras. ¿Cómo caracteriza el autor a cada una? ¿Cuál es el problema que se plantea? ¿Cómo podría solucionarse?

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……