

Laboratorio Nro. 5

Programación Dinámica

Objetivos

...

Consideraciones iniciales

Leer la Guía



Antes de comenzar a resolver el presente laboratorio, leer la **“Guía Metodológica para la realización y entrega de laboratorios de Estructura de Datos y Algoritmos”** que les orientará sobre los requisitos de entrega para este y todos los laboratorios, las rúbricas de calificación, el desarrollo de procedimientos, entre otros aspectos importantes.

Registrar Reclamos



En caso de tener **algún comentario** sobre la nota recibida en este u otro laboratorio, pueden **enviarlo** a través de <http://bit.ly/2g4TTKf>, el cual será atendido en la menor brevedad posible.

Traducción de Ejercicios



En el GitHub del docente, encontrarán la traducción al español de los enunciados de los Ejercicios en Línea.

**Visualización de
Calificaciones**

A través de **Eafit Interactiva** encontrarán un **enlace** que les permitirá **ver un registro de las calificaciones** que **emite el docente** para cada taller de laboratorio y según las rubricas expuestas. **Véase sección 3, numeral 3.8.**

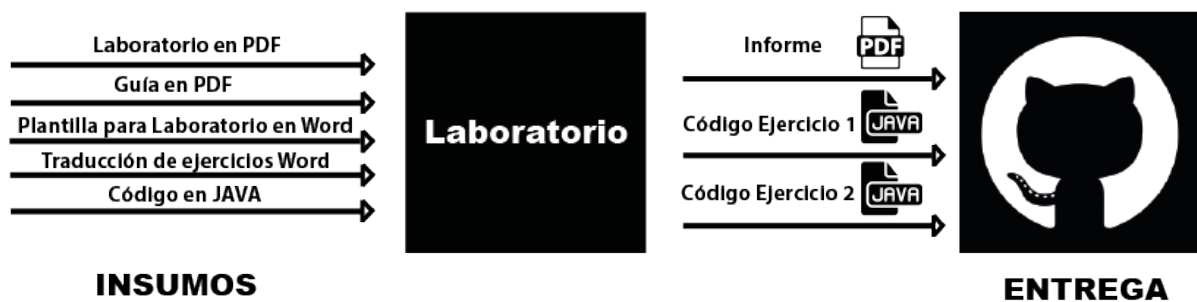
GitHub

1. Crear un repositorio en su cuenta de GitHub con el nombre `st0247-suCodigoAqui`. 2. Crear una carpeta dentro de ese repositorio con el nombre `laboratorios`. 3. Dentro de la carpeta `laboratorio`, crear una carpeta con nombre `lab05`. 4. Dentro de la carpeta `lab04`, crear tres carpetas: `informe`, `codigo` y `ejercicioEnLinea`. 5. Subir el informe pdf a la carpeta `infome`, el código del ejercicio 1 a la carpeta `codigo` y el código del ejercicio en línea a la carpeta `ejercicioEnLinea`. Así:

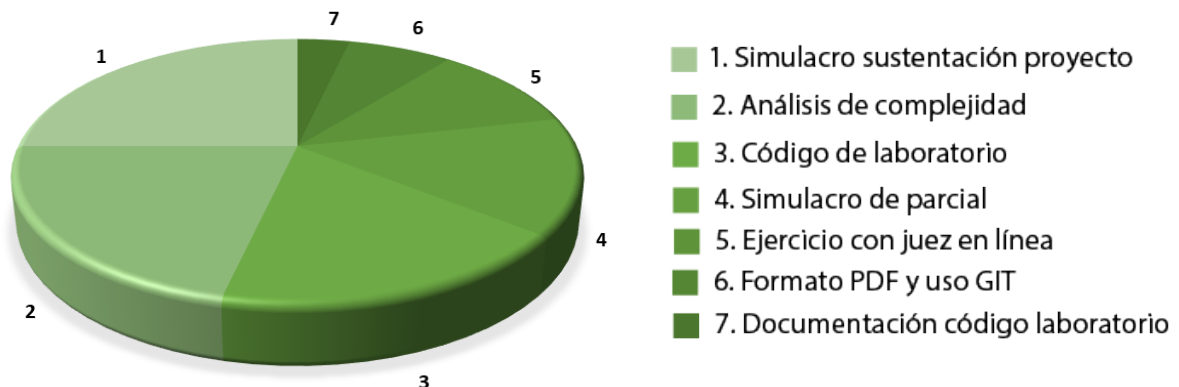
```
st0247-suCodigoAqui
  laboratorios
    lab01
      informe
      codigo
      ejercicioEnLinea
    lab02
      ...
```

Intercambio de archivos

Los archivos que **ustedes deben entregar** al docente son: **un archivo PDF** con el informe de laboratorio usando la plantilla definida, y **dos códigos**, uno con la solución al numeral 1 y otro al numeral 2 del presente. Todo lo anterior se entrega en **GitHub**.



Porcentajes y criterios de evaluación para el laboratorio



Resolver Ejercicios

1. Códigos para entregar en GitHub:



En la vida real, la documentación del software hace parte de muchos estándares de calidad como CMMI e ISO/IEC 9126



Véase Guía *en Sección 3, numeral 3.4*



Código de laboratorio en *GitHub*. Véase Guía en *Sección 4, numeral 4.24*



Documentación en *HTML*



No se reciben archivos en *.RAR* ni en *.ZIP*



En la vida real, el problema del agente viajero se aplica para la construcción de circuitos electrónicos, recolectar monedas de teléfonos de monedas, entre otras. Léase más en <http://bit.ly/2i9JdIV>

1.1 Implemente el algoritmo de Held-Karp, también conocido como algoritmo de programación dinámica para el agente viajero.

2) Ejercicios en línea sin documentación HTML en GitHub:



Véase Guía en **Sección 3, numeral 3.3**



No entregar
documentación **HTML**



Entregar un archivo
en **.JAVA**



No se reciben archivos
en **.PDF**



Resolver los problemas
de **CodingBat** usando
Recursión



Código del ejercicio en línea
en **GitHub**. Véase Guía en
Sección 4, numeral 4.24

2.1 Resuelvan el siguiente problema usando programación dinámica:

...

3) Simulacro de preguntas de sustentación de Proyectos



Véase Guía en **Sección 3,**
Numeral 3.5



Entregar informe
de laboratorio en **PDF**



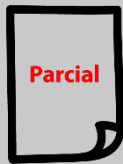
Usen la **plantilla** para
responder laboratorios



No apliquen Normas
Icontec para esto

3.1 ...

4) Simulacro de Parcial en el informe PDF



Para este simulacro, agreguen ***sus respuestas*** en el informe PDF.



El día del Parcial no tendrán computador, JAVA o acceso a internet.



PISTA 1: Véase *Guía en Sección 4, Numeral 4.18* “Respuestas del Quiz”



PISTA 2: Lean las diapositivas tituladas “*Data Structures II: Dynamic programing*”, encontrarán la mayoría de las respuestas

1. ...

5. [Ejercicio Opcional] Lectura recomendada



"Quienes se preparan para el ejercicio de una profesión requieren la adquisición de competencias que necesariamente se sustentan en procesos comunicativos. Así cuando se entrevista a un ingeniero recién egresado para un empleo, una buena parte de sus posibilidades radica en su capacidad de comunicación; pero se ha observado que esta es una de sus principales debilidades..."

Tomado de <http://bit.ly/2gJKzJD>



Véase Guía en **Sección 3, numeral 3.6 y 4.20** de la Guía Metodológica, “Lectura recomendada” y “Ejemplo para realización de actividades de las Lecturas Recomendadas”, respectivamente

Posterior a la lectura del “**R.C.T Lee et al., Introducción al análisis y diseño de Algoritmos. Capítulo ¿???. Páginas ¿??.**”, realicen las siguientes actividades que les permitirán sumar puntos adicionales:

- a) Escriban un resumen de la lectura que tenga una longitud de 100 a 150 palabras



PISTA 1: En el siguiente enlace, unos consejos de cómo hacer un buen resumen <http://bit.ly/2knU3Pv>



PISTA 2: [Aquí](#) le explican cómo contar el número de palabras en Microsoft Word

- b) Hagan un mapa conceptual que destaque los principales elementos teóricos.



PISTA: Para que hagan el mapa conceptual se recomiendan herramientas como las que encuentran en <https://cacoo.com/> o <https://www.mindmup.com/#m:new-a-1437527273469>



NOTA 1: Si desean una lectura adicional en español, consideren la siguiente: “**John Hopcroft et al., Estructuras de Datos y Algoritmos, Sección 10.3. 1983**”, que encuentran en biblioteca.



NOTA 2: Estas respuestas también deben incluirlas en el informe PDF

6. [Ejercicio Opcional] Trabajo en Equipo y Progreso Gradual



El trabajo en equipo es una exigencia actual del mercado. "Mientras algunos medios retratan la programación como un trabajo solitario, la realidad es que requiere de mucha comunicación y trabajo con otros. Si trabajas para una compañía, serás parte de un equipo de desarrollo y esperarán que te comuniques y trabajes bien con otras personas"

Tomado de <http://bit.ly/1B6hUDp>



Véase Guía en **Sección 3, numeral 3.7** y **Sección 4, numerales 4.21, 4.22 y 4.23** de la Guía Metodológica

- a) Entreguen copia de todas las actas de reunión usando el tablero Kanban, con fecha, hora e integrantes que participaron



PISTA: Véase **Guía en Sección 4, Numeral 4.21** "Ejemplo de cómo hacer actas de trabajo en equipo usando Tablero Kanban"

- b) Entreguen el reporte de *git*, *svn* o *mercurial* con los cambios en el código y quién hizo cada cambio, con fecha, hora e integrantes que participaron



PISTA: Véase Guía en Sección 4, Numeral 4.23 "Cómo generar el historial de cambios en el código de un repositorio que está en svn"

”

- c) Entreguen el reporte de cambios del informe de laboratorio que se genera Google docs o herramientas similares



PISTA: Véase Guía en Sección 4, Numeral 4.22 "Cómo ver el historial de revisión de un archivo en Google Docs"



NOTA: Estas respuestas también deben incluirlas en el informe PDF

Resumen de ejercicios a resolver

1.1 ...

4. Simulacro de Parcial

5. Lectura recomendada **[Ejercicio Opcional]**

6. Trabajo en Equipo y Proceso Gradual **[Ejercicio Opcional]**