



Universidad del Istmo de Guatemala  
Facultad de Ingenieria  
Ing. en Sistemas  
Informatica 1  
Prof. Ernesto Rodriguez - erodriguez@unis.edu.gt

---

## Hoja de trabajo #1

Fecha de entrega: 25 de Julio, 2018 - 11:59pm

---

*Instrucciones: Resolver cada uno de los ejercicios siguiendo sus respectivas instrucciones. El trabajo debe ser entregado a traves de Github, en su repositorio del curso, colocado en una carpeta llamada "Hoja de trabajo 1". Al menos que la pregunta indique diferente, todas las respuestas a preguntas escritas deben presentarse en un documento formato pdf, el cual haya sido generado mediante Latex.*

### Iniciación

1. Si no la tiene, crear una cuenta en Github.
2. Crear un repositorio[2] para los deberes correspondientes a este curso en Github.
3. Enviar su usuario de Github y link de su repositorio al profesor erodriguez@unis.edu.gt.
4. Tomar en cuenta que esta hoja (y las hojas restantes del curso) deben ser entregadas mediante un commit en el repositorio creado en el articulo 2 de Github. En la fecha y hora de entrega, el profesor copiara el estado actual del repositorio y calificara ese material. Por favor asegurarse que todos los cambios esten *empujados* (`git push`) en el repositorio antes de la fecha de entrega. Esto se puede revisar accediendo el sitio Github y ver que todos los archivos necesarios esten presentes en el repositorio.
5. El "Ejercicio #1" y "Ejercicio #4" le ayudaran a publicar correctamente el trabajo en Github. Si aun tiene dudas, por favor contactar al profesor.
6. Para hacer esta hoja de trabajo, necesitara Latex[3]. Para trabajar con archivos Latex (\*.tex) se recomienda utilizar un editor. La extension de Latex para *Visual Studio Code* es excelente, pero también se puede utilizar un programa como Texmaker. Asegurese de configurar Latex de manera que pueda producir archivos *PDF*.
7. El Wiki de Latex[4] es un buen lugar para consultar el formato.

### Ejercicio #1 (5%)

1. Si no lo ha hecho, crear un repositorio en Github para alojar las hojas de trabajo.

2. Luego de haber creado un repositorio, debe clonar el repositorio a una carpeta local mediante el comando `git clone [url]` el cual puede ser ejecutado desde la terminal (Linux y Mac) o desde la consola *Git Bash* (en Windows). Para mayor facilidad, utilizar un url tipo *https*. Sin embargo para mayor seguridad y conveniencia se recomienda usar *ssh*[1]. Como ejemplo, este comando clonaria el repositorio de este curso:

```
git clone https://github.com/netogallo/informatica-2018-2019
```

3. Luego de haber clonado el repositorio, se creara una carpeta con el nombre del repositorio dentro de la carpeta donde se haya ejecutado el comando anterior.

## Ejercicio #2: Abstracción (50%)

Consideremos un dado de seis caras colocado en una superficie:

- La cara opuesta a la superficie muestra el *numero actual* en que se encuentra el dado.
- El dado puede cambiar su *numero actual* rotando 90 grados en cualquiera de sus 4 lados. La imagen muestra el dado cambiando su *nuero actual* de “2” a “6” rotando sobre la cara “1”.



Su deber es modelar el dado y sus transiciones de estado como un *grafo*. Para ello debe indicar:

1. El *conjunto* de nodos del grafo.
2. El *conjunto* de vertices del grafo.

**Pista:** Por lo general, los vertices corresponden a transiciones validas que pueden ocurrir en el objeto que esta siendo modelado.

## Ejercicio #3 (40%)

Indique como puede utilizar el *grafo* creado en el paso anterior para simular un lanzamiento de dados. Supongamos, que el dado siempre inicia en el *numero actual* “1”:

- ¿Que estructura de datos podria representar un lanzamiento de dados?
- ¿Que algoritmo podriamos utilizar para generar dicha estructura?
- ¿Como nos aseguramos que ese algoritmo siempre produce un resultado?

## Ejercicio #4 (5%)

Luego de haber terminado los ejercicios anteriores, su trabajo debe ser publicado a github. Para publicar a Github utilizando Git se procede de la siguiente manera:

1. El comando `git status` listara todos los cambios pendientes por agregar al repositorio. Si no ha agregado ningun archivo al repositorio, listara todas las carpetas y el archivo *.gitignore* dentro de la carpeta del proyecto.
2. El comando `git add [archivos]` permite agregar archivos a la siguiente revision de git. Como atajo, puede ejecutar `git add .` para agregar todos los cambios que hay en el repositorio.

3. El comando `git status` ahora indicara los archivos nuevos que hay en el repositorio. Si su archivo `.gitignore` esta formateado correctamente, solo deberian agregarse:
- El archivo `.gitignore`
  - El archivo `pdf` correspondiente a esta hoja de trabajo
  - El archivo `tex` correspondiente a esta hoja de trabajo
  - Un posible archivo o carpeta de configuración del editor. El caso de “Visual Studio Code” sera una carpeta llamada `.vscode`.
4. Si el paso anterior muestra archivos extra (`*.aux`, `*.log`, `*.out`, ect...), debera componer su archivo `.gitignore`. Para hacerlo, primero debe ejecutar el comando `git reset HEAD .`, el cual quitara todos los archivos de la lista de cambios. Luego modifique su archivo `.gitignore` adecuadamente y vuelva a ejecutar `git add .`. Repetir hasta que la lista de archivos nuevos corresponda a la indicada en el paso #3.

## References

- [1] Github. Connecting to github with ssh. <https://help.github.com/articles/connecting-to-github-with-ssh/>.
- [2] Github. Creating a new repository. <https://help.github.com/articles/creating-a-new-repository/>.
- [3] Latex Project. Latex. <https://www.latex-project.org/>.
- [4] Wikibooks. Latex. <https://en.wikibooks.org/wiki/LaTeX>.