



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

---

## Proyecto Final: Fractales

Fecha de entrega: 01 de Noviembre, 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 "Proyecto Final". 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.*

A continuación se presenta el proyecto final. El objetivo de este proyecto es familiarizar al estudiante con la utilización de la programación funcional para crear programas interactivos. El objetivo es que el estudiante construya un programa que dibuje fractales.

Para este proyecto pude utilizar como machote ya sea el codigo de esta carpeta o el proyecto ubicado en <https://ellie-app.com/3tH24kjVKPJal>. Si utiliza el proyecto del link, por favor cree su propia copia en vez de modificar el codigo del link directamente para que sus comañeros puedan obtener una copia sin modificación si lo desean.

El proyecto se puede llevar a cabo de forma individual o en pareja. Si se realiza en pareja, por favor indicar claramente quien es su pareja y asegurarse que exista una copia del proyecto en el repositorio de ambos. Este proyecto tiene como valor 15 puntos de los 40 que conforman el examen final. Adicionalmente, existe la oportunidad de 15 puntos extra de los cuales 5 seran para el examen final y 10 para recuperar nota en los parciales anteriores.

Aparte de la entrega final, el dia 01 de noviembre, tendra que presentarle el codigo al profesor durante la clase. El profesor tiene derecho de hacerle preguntas sobre cualquier parte del codigo a cualquiera de los integrantes del grupo. La nota final del proyecto es individual (no en pareja) y dependera de la habilidad de cada integrante en responder las preguntas que el profesor le haga sobre el codigo.

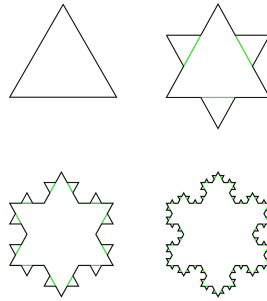
## Iniciación

Esta sección solo importa si va utilizar este repositorio como machote. Si utilizara el codigo ubicado en Ellie, puede ignorar esta sección. Para utilizar este repositorio como machote, debe seguir los siguientes pasos:

1. Copiar el contenido de esta carpeta a su carpeta llamada "Proyecto Final"
2. Navegar a la carpeta "Proyecto Final" en la terminal
3. Ejecutar `npm install` en la terminal

4. Ejecutar `npm run build` en la terminal
5. Ejecutar `node_modules/.bin/elm reactor` en la terminal
6. Navegar a `http://localhost:8000/index.html` en su navegador y verificar que el código funciona

## Copo de Nieve de Koch (5pts)



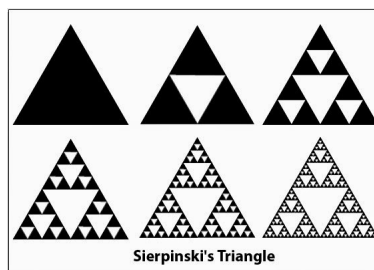
El copo de nieve de Koch es un polígono definido recursivamente. Dado un polígono (un triángulo por ejemplo), se puede generar el siguiente polígono así:

1. Dividir cada línea del polígono en 3 partes exactas.
2. Crear un triángulo nuevo en el espacio abarcado por la línea del medio
  - La longitud de cada línea del triángulo es la misma que la línea del medio.

Su tarea consiste en definir una función llamada `snowflake` :  $\mathbb{N} \rightarrow \text{List } (\mathbb{R}, \mathbb{R})$ . Esta función debe aceptar un número natural indicando la cantidad de veces que se repetirá la recursión. Si el número es 1, el resultado debe ser un triángulo, similar al valor `fractal` que se define en la línea 11 de “Main.elm”.

Si desea visualizar su fractal para verificar si funciona correctamente, puede modificar la línea 36 de “Main.elm” colocando una llamada a su función con algún número en vez de `fractal`. Por ejemplo: `poligono ctx = dibujar (snowflake 5) ctx`. Debe volver a ejecutar `npm run build` y los resultados se pueden visualizar en `http://localhost:8000/index.html` al correr `node_modules/.bin/elm reactor`

## Triángulo de Sierpinski (10pts)



El triángulo de Sierpinski es otro fractal generado recursivamente. El caso base consiste en un triángulo regular. Luego, cada paso recursivo divide cada uno de los triángulos que conforman el triángulo mediante otro triángulo que cada lado del triángulo está ubicado en la mitad de cada una de las líneas del triángulo anterior.

Su tarea consiste en definir la función `sierpinski` :  $\mathbb{N} \rightarrow \text{List}(\text{List}(\mathbb{R}, \mathbb{R}))$  la cual debe aceptar un número natural como parámetro y generar una lista de triángulos que conforman al triángulo de Sierpinski luego de aplicar la regla recursiva el número de veces indicado por el parámetro numérico.

También debe definir la función `dibujarTriangulos` :  $\text{List}(\text{List}(\mathbb{R}, \mathbb{R})) \rightarrow \text{Canvas.Commands} \rightarrow \text{Canvas.Commands}$ . Esta función debe utilizar la función `dibujar` que ya se definió en “Main.elm” en la línea 19. Para dibujar todos los triángulos en la lista que se le ha dado. Para ello se debe:

1. Llamar la función `dibujar` con cada uno de los triángulos en la lista
2. Cada llamada produce como resultado un contexto, el cual se utiliza como parámetro para la siguiente llamada.
3. Para la primera llamada, utilizar el contexto recibido como parámetro

Meida vez hayan sido definidas estas funciones, puede dibujar su triángulo de Sierpinski modificando la línea 45 de “Main.elm” con `poligono ctx = dibujarTriangulos (sierpinski 5) ctx`.

## Extras (15pts)

Se otorgan hasta 15 puntos extras por artículos adicionales en la entrega que estén relacionados al trabajo. Algunos ejemplos podrían ser:

- Controles para decidir que fractal se debe visualizar
- Controles para controlar el número de recursiones que se realizarán
- Utilizar css para mejorar la apariencia visual del programa
- Inclusión de otros fractales
- Utilizar color diferente para cada recursión dibujada para mejor visualización del fractal.
- Opción de dibujar el fractal de forma probabilística. Ie. a los puntos que conforman las esquinas del fractal se les modifica la posición ligeramente mediante números aleatorios.
- Interactividad con el mouse