

# ORIGAMI: Herramienta para apoyar el Aprendizaje de Programación Estructurada basada en Diagramas de Flujo



Juan Ku Quintana

[juankuq@gmail.com](mailto:juankuq@gmail.com)

Víctor Rodríguez Cámara

[victor.fmat@gmail.com](mailto:victor.fmat@gmail.com)

Edgar Cambranes Martínez

[edgar.cambranes@uady.mx](mailto:edgar.cambranes@uady.mx)



SACULTAD DE  
MATEMÁTICAS

# Índice

1

Introducción

2

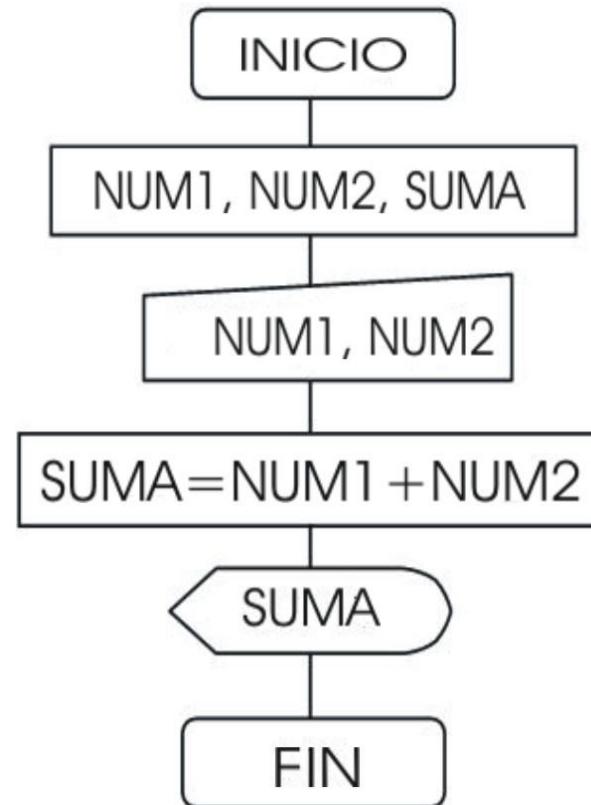
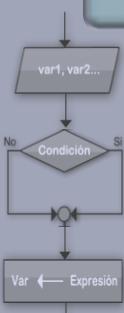
Metodología

3

Vista general

4

Resultados



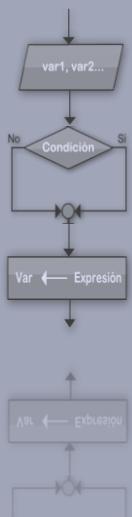
# Problemática

1

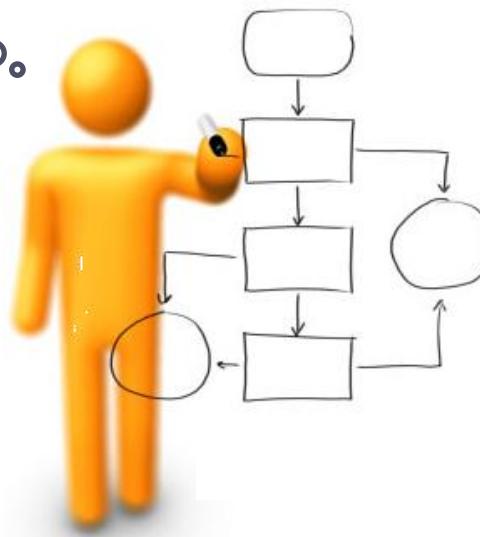
2

3

4



Generación de pensamiento lógico



# Antecedentes

1

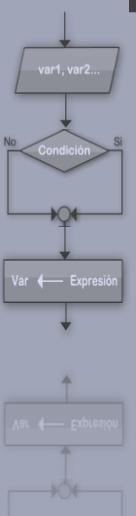
➤ Según David A. Scanlan “Los métodos gráficos deben de ser fuertemente considerados cuando se enseñan algoritmos relativamente complejos”

2

➤ Ackermann encontró que hay una mejoría del 27% cuando se utilizó una herramienta de software que ayude a la construcción de diagramas de flujo

3

➤ Facultad de matemáticas.- trabajo colegiado entre profesores, ejercicios, herramientas, secuencia de temas y formas de evaluación.



# Herramientas Actuales

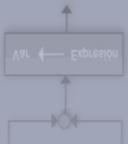
- No son diseñadas hacia el aprendizaje  
Toman en cuenta solamente la funcionalidad principal
- No siguen un diseño centrado en el usuario  
Características psicológicas, cognitivas y ambientales
- Débil usabilidad  
Facilidad de uso, de aprendizaje, intuitiva

1

2

3

4



# ORIGAMI: una propuesta de solución

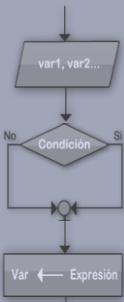
1

- Apoyar al estudiante en el aprendizaje de PE a través de diagramas de flujo

2

3

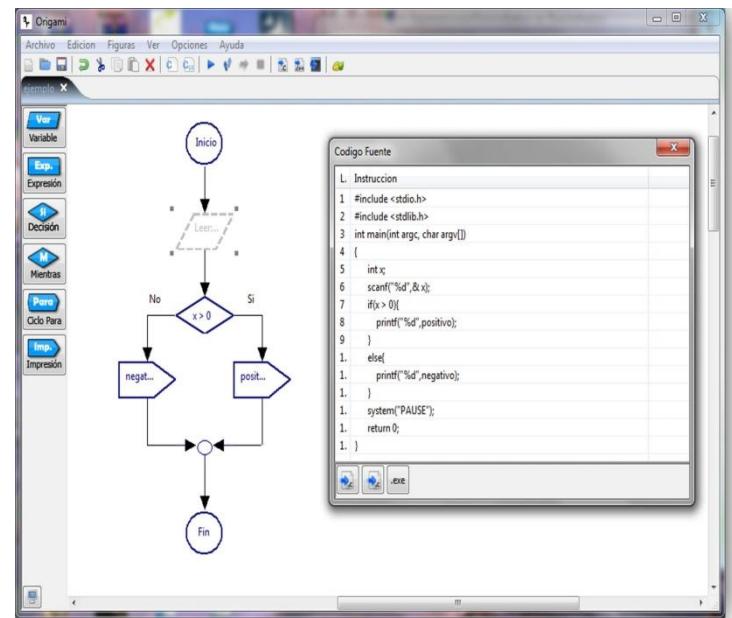
4



- Metodología centrada en el usuario

- Facilidad de uso

- Funcionalidades de apoyo



# Metodología

1

2

3

4



- Aplicación de pruebas
- Retroalimentación

Retroalimentación

Análisis

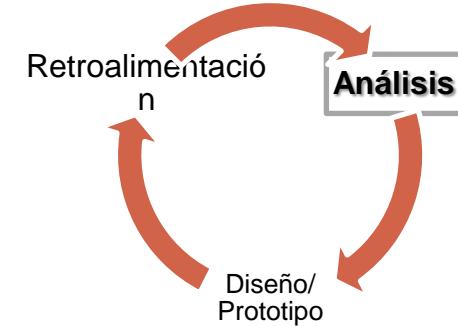
Diseño  
Centrado en  
el Usuario

Diseño/  
Prototipo

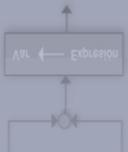
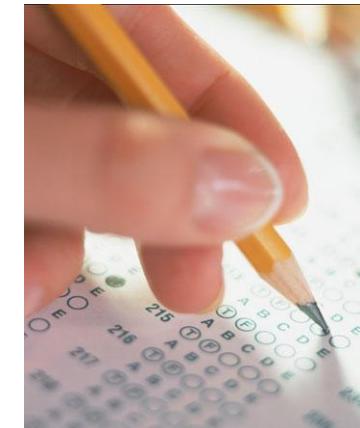
- Recorridos cognitivos
- Diseño de pruebas de uso
- Revisión de heurísticas

- Perfiles
- Personas
- Escenarios

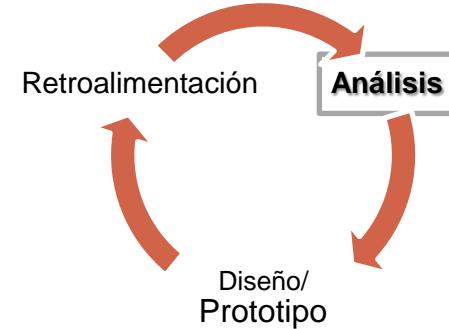
# Análisis



- 1 ➤ Heterogeneidad en el grupo.
- 2 ➤ Diferentes perfiles
- 3 ➤ Encuesta a profesores
  - Funcionalidades consideradas
- 4 ➤ Metodología utilizada para la enseñanza de lógica de programación
  - DF vs Pseudocódigo

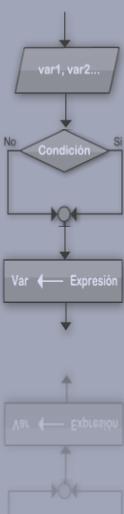
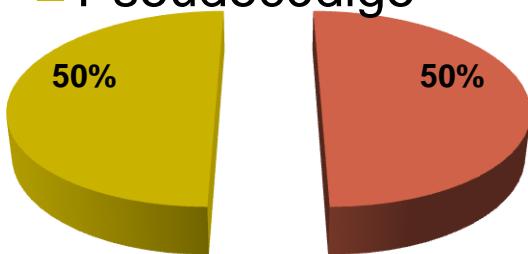


# Resultados de la encuesta



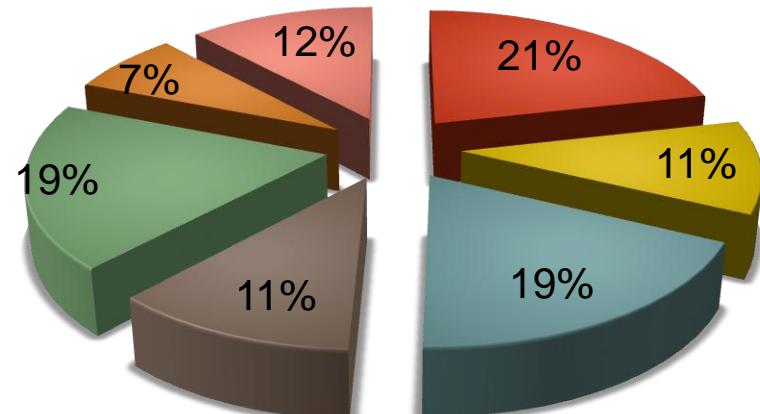
## DF vs Pseudocódigo

- Diagramas de flujo
- Pseudocódigo

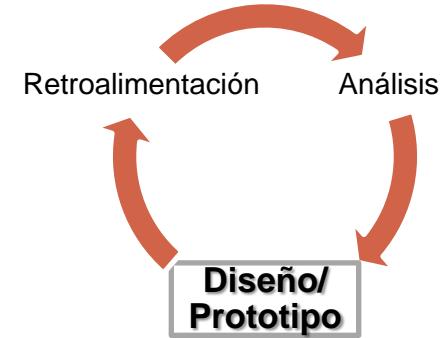


## Funcionalidades

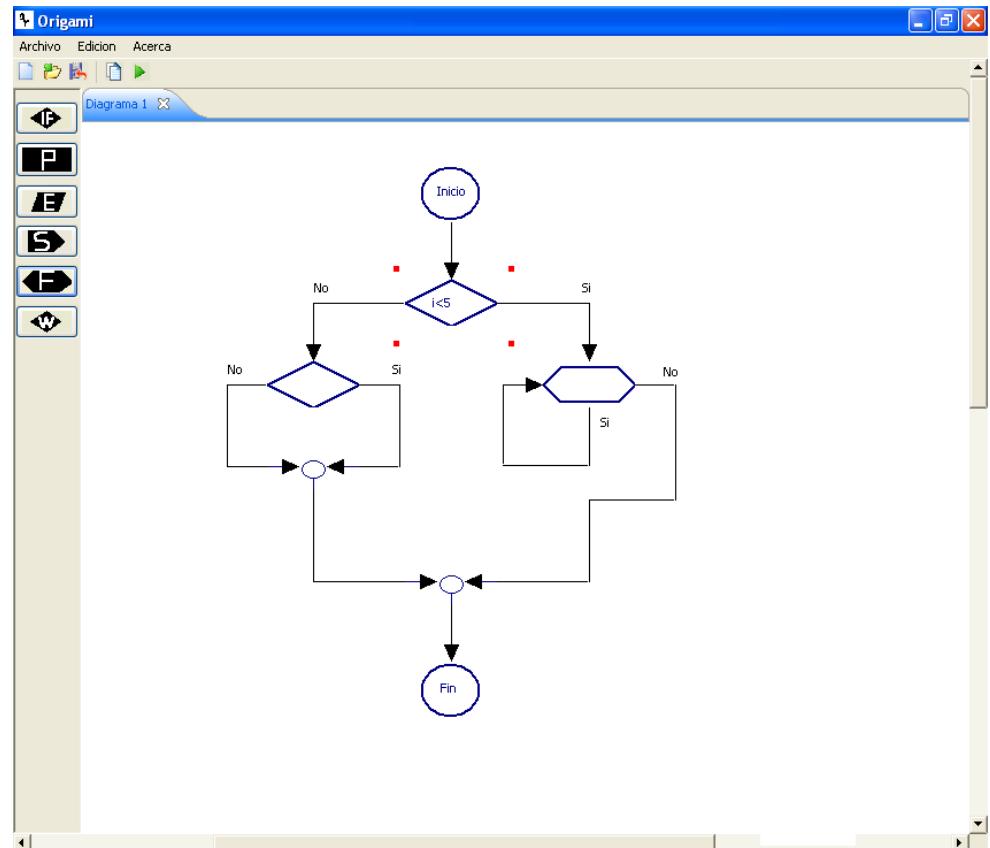
- Paso a Paso
- Guardar diagrama en imagen
- Diagrama a Pseudo código
- Depurador
- Pseudo código a diagrama
- Subprograma/Función

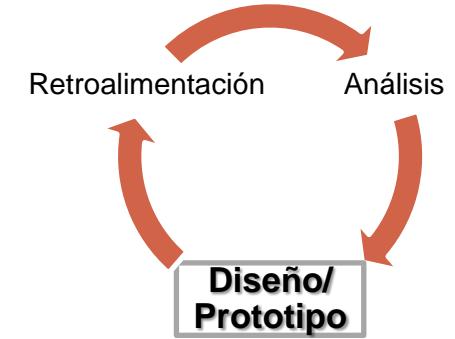


# Diseño/Prototipo



- Aspectos de interfaces conocidas
- Agregar instrucciones
- Semi-dirijida del diagrama.





# Diseño/Prototipo

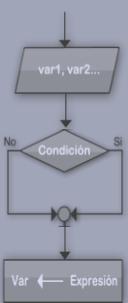
## Prueba de usabilidad:

1

2

3

4



### ➤ Objetivo

Conocer los aspectos de interacción que puedan dificultar la comprensión y aprendizaje de la herramienta.

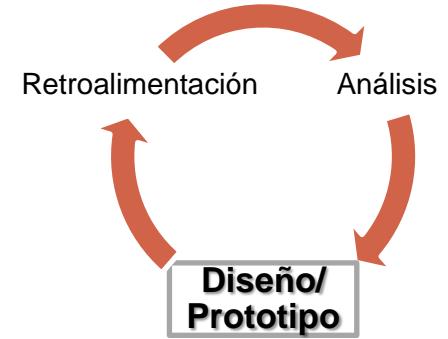
### ➤ Metodología

Aplicada a estudiantes de la Facultad de Matemáticas de la UADY en Mérida, Yucatán.

### ➤ Resultados

Retroalimentación interfaz.

# Actividad - Estudiantes



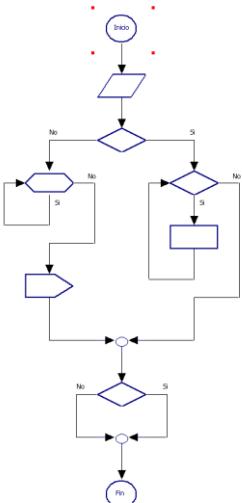
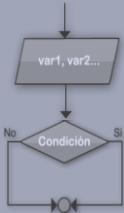
- 1) Desarrollar un diagrama

1

2

3

4



- 2) Agregar instrucciones
- 3) Eliminar un símbolo

Created using Work

Created using Work

Test - 1

Test - 2

Created using Work

Created using Work

Test - 3

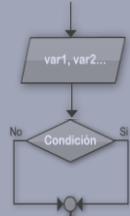
Test - 4

1

2

3

4

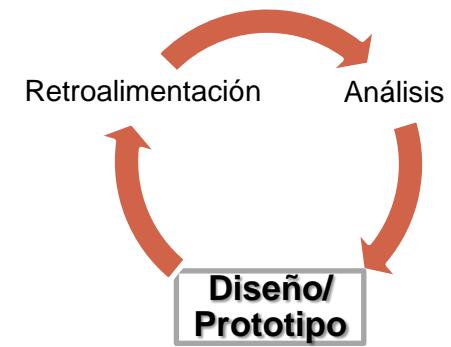


# Actividad - Estudiantes

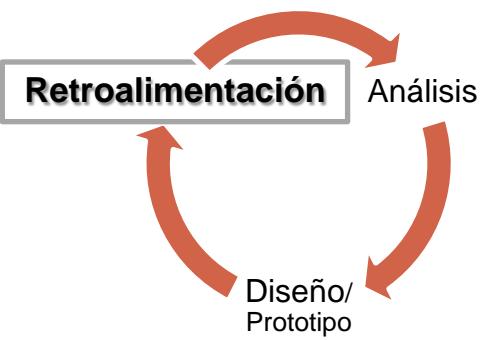
Contestar las siguientes preguntas:

Mala	Deficiente	Regular	Buena	Excelente
------	------------	---------	-------	-----------

- ¿Se te hizo fácil usarlo?
- ¿Se te hizo fácil realizar la actividad (factorial) con la ayuda de Origami?
- ¿Identificaste fácilmente los símbolos (decisión, para, mientras, proceso, entrada, salida)?
- ¿Ocurrió algún error mientras la usabas?



# Retroalimentación



## Resultados de la prueba

1

➤ Retroalimentación de la interfaz de usuario

- Mejorar iconografía
- Nombres en las figuras del diagrama

2

➤ Mejorar accesibilidad a las funciones

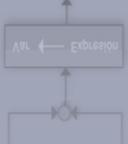
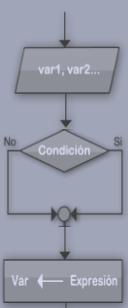
- Agregar instrucción
- Funciones de diseño(cortar, copiar, pegar)

3

➤ Desarrollar nuevas funcionalidades

- Paso a paso
- Exportar código C/C++

4



# Características de Origami

1

- Accesos intuitivos

2

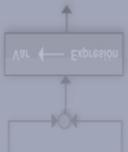
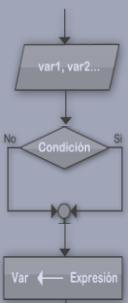
- Ejecución del diagrama de flujo

3

- Generar código C/C++

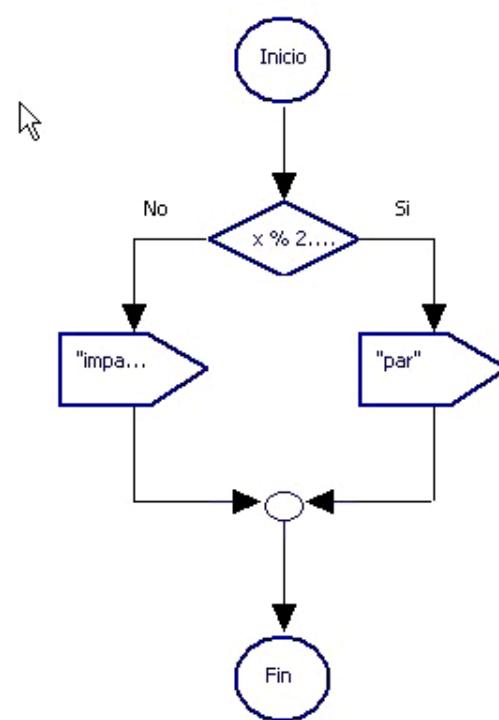
4

- Almacena información sobre la construcción del diagrama de flujo





\*g

**Var**  
Variable**Exp.**  
Expresión**Si**  
Decisión**M**  
Mientras**Para**  
Ciclo Para**Imp.**  
Impresión

# Resultados Iniciales

- ❖ Actualmente la herramienta esta siendo usada por alumnos de primer semestre de la facultad de matemáticas en la materia de programación estructurada.

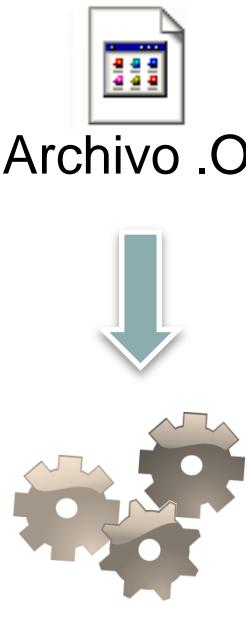
2008	
Total de Estudiantes del Curso de FP	138
Promedio general del Primer Parcial	46.37
Estudiantes Aprobados	54 (39.13 %)

2009	
Total de Estudiantes del Curso de FP	131
Promedio general del Primer Parcial	67.94
Estudiantes Aprobados	90 (68.70%)

*Variables involucradas:*

Carga Académica, uso de herramientas de programación, trabajo colegiado de profesores

# Resultados Iniciales



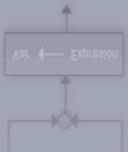
- ❖ Tiempo total que estuvo abierto el archivo
- ❖ Bloques utilizados en el diagrama
- ❖ Errores de compilación
  - Inicio nuevo diagrama
  - Inicio
  - Fin
- ❖ Aspectos comunes que dificultan una tarea en específico.
  - YA se agregó una nueva figura al diagrama
  - La figura agregada es de tipo "si".
  - Inicio
  - Entrada (Leer: int x)
  - If(x%2==0) {
  - Else
  - }
  - Fin
- ❖ Análisis de avance individual o grupal

/Ec - Error en la compilación: <ERROR>  
main.cpp: In function 'int main (int char\*)'  
main.cpp:7: error: 'i' undeclared (first use this function)  
main.cpp:7: error: (Each undeclared identifier is  
Reported only once for each function it appears in.)

Tiempo total = 15:52:19 - 15:57:38

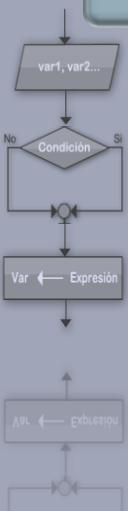
# Trabajo Futuro

- Refactorización
  - Código entendible
  - Fácil de mantener
- Desarrollo de herramienta de análisis de la construcción de DF
  - Analizar el avance de los estudiantes y modelo lógico-computacional
- Distribución
  - Aportación de FMAT a la comunidad.
- Trabajo en conjunto con otras herramientas
  - Objetos de aprendizaje
  - Verificador de código



# RESUMEN

- Problemática
  - Dificultad para generar un pensamiento lógico
  - Abstracción de algoritmos
- Metodología DCU
  - Técnicas y herramientas
  - Facilitar el uso de la herramienta
- Desarrollo y evaluación de la herramienta
  - Retroalimentación interfaz (Interacción)
- Uso en la asignatura de Programación Estructurada
  - Dinámica en la clase
  - Almacenar información



1

2

3

4

# ORIGAMI: Herramienta Para Apoyar El Aprendizaje De Programación Estructurada Basada En Diagramas De Flujo



Juan Ku Quintana

[juankuq@gmail.com](mailto:juankuq@gmail.com)

Víctor Rodríguez Cámara

[victor.fmat@gmail.com](mailto:victor.fmat@gmail.com)

Edgar Cambranes Martínez

[edgar.cambranes@uady.mx](mailto:edgar.cambranes@uady.mx)

