

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

Juan Ku Quintana

juankuq@gmail.com

Víctor Rodríguez Cámara

victor.fmat@gmail.com

Edgar Cambranes Martínez

edgar.cambranes@uady.mx



UADY

FACULTAD DE
MATEMÁTICAS

"Luz, Ciencia y Verdad"

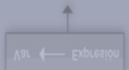
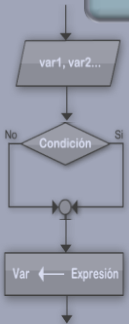
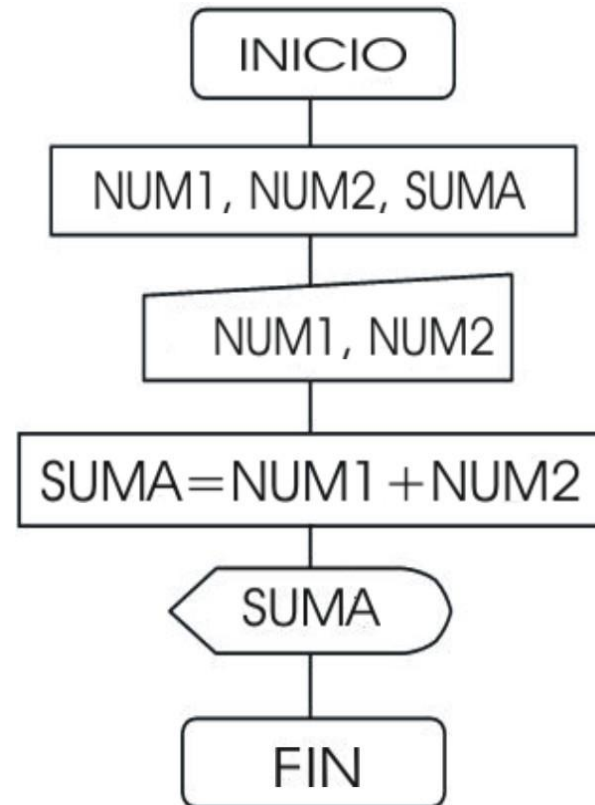
„Лгш' Сгелсгс ʎ ʎелсгс“



UNIVERSIDAD
AUTÓNOMA DE

Índice

- 1 Introducción
- 2 Metodología
- 3 Vista general
- 4 Resultados



Problemática

1



Generación de pensamiento lógico

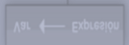
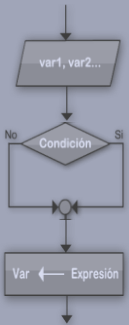
Abstracción de los algoritmos

Dificultad para comprobar algoritmos y darle seguimiento a algoritmos complejos

2

3

4



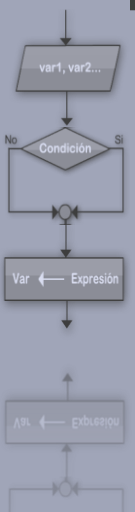
Antecedentes

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

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



Herramientas Actuales

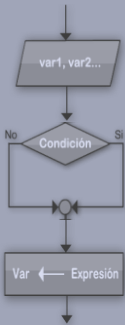
- No son diseñadas hacia el aprendizaje
Toman en cuenta solamente la funcionalidad principal
- No siguen un diseño centrado en el usuario
Características psicológicas, cognitivas y ambientales
- Débil usabilidad
Facilidad de uso, de aprendizaje, intuitiva

1

2

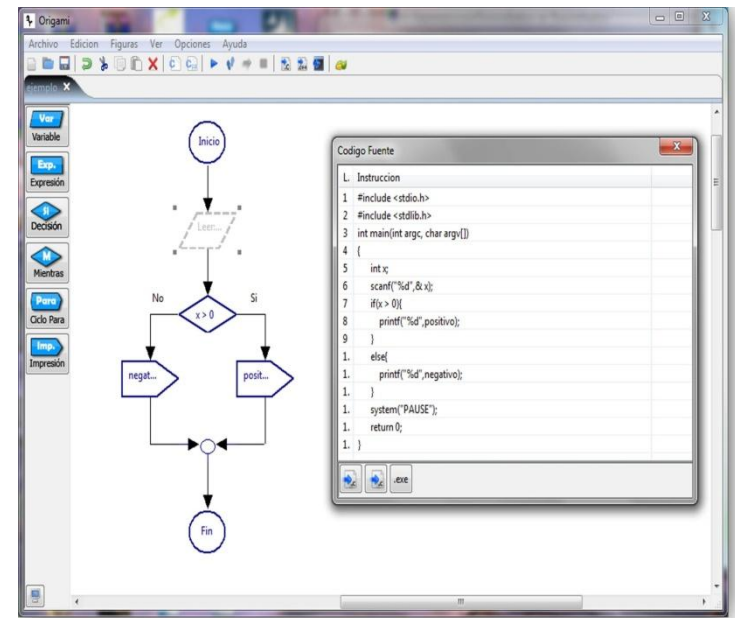
3

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

- Perfiles
- Personas
- Escenarios

Retroalimentación

Análisis

Diseño
Centrado en
el Usuario

Diseño/
Prototipo

- Aplicación de pruebas
- Retroalimentación

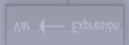
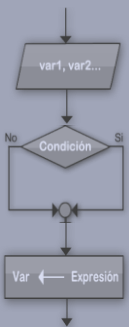
- Recorridos cognitivos
- Diseño de pruebas de uso
- Revisión de heurísticas

1

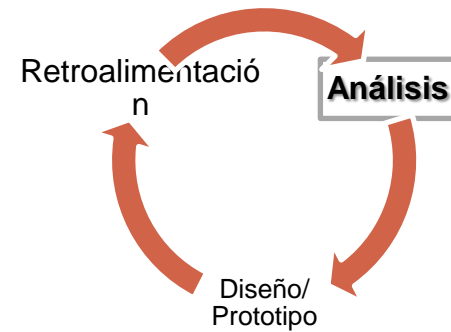
2

3

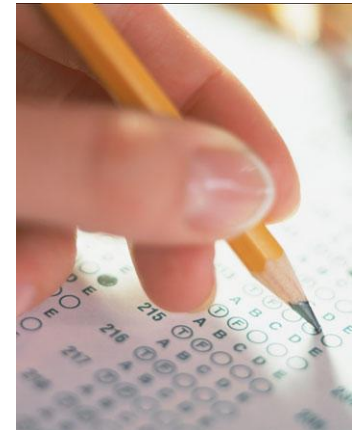
4



Análisis



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

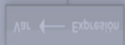
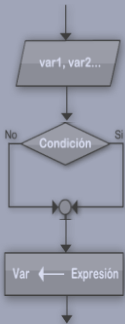


1

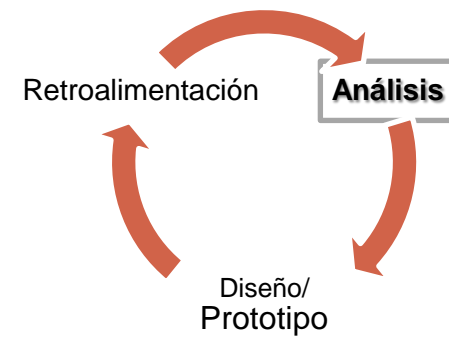
2

3

4



Resultados de la encuesta



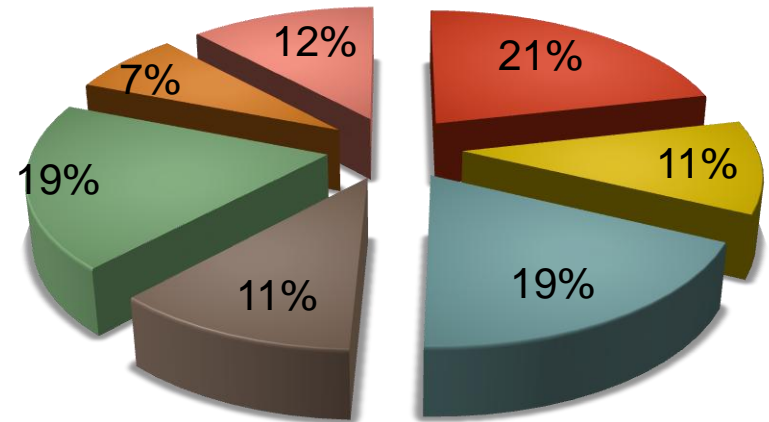
DF vs Pseudocódigo

- Diagramas de flujo
- Pseudocódigo



Funcionalidades

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

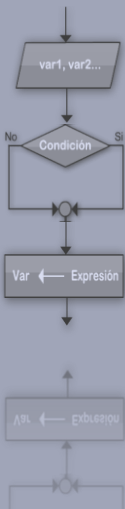


1

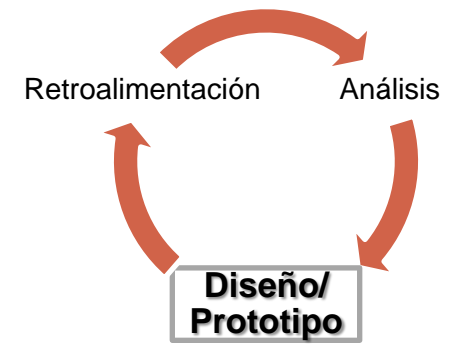
2

3

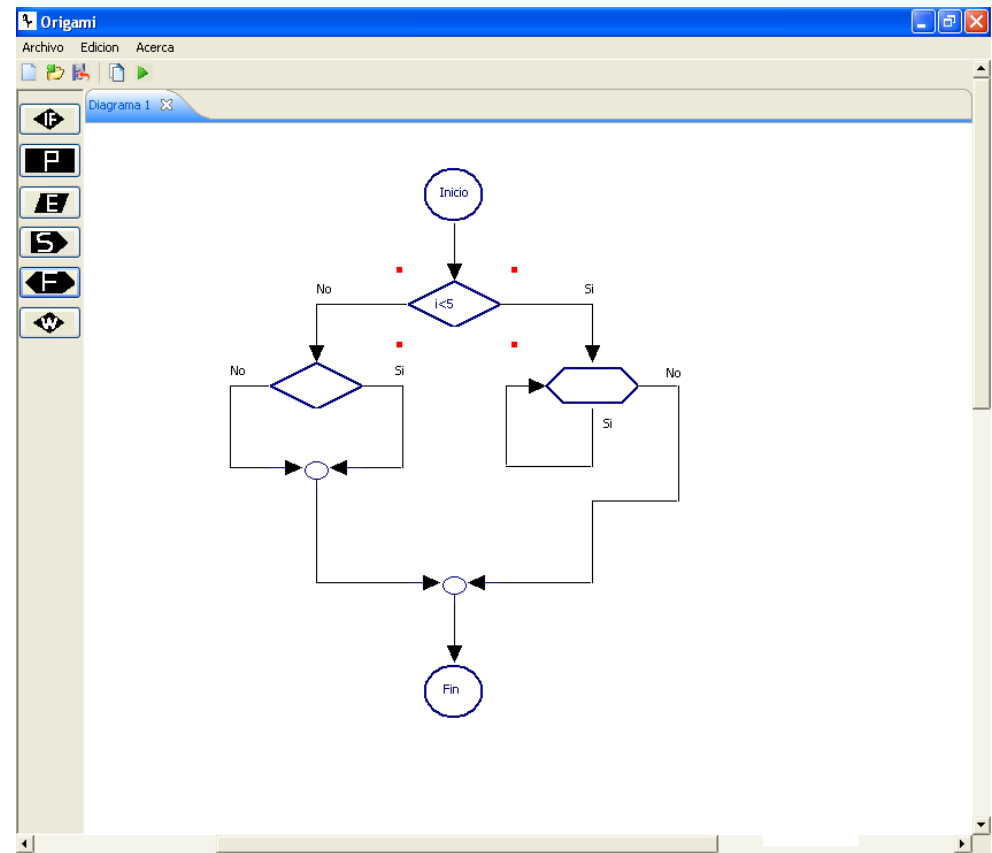
4



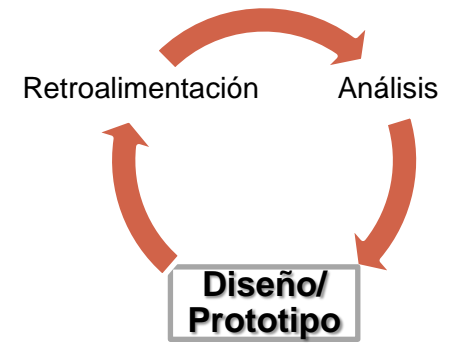
Diseño/Prototipo



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



Diseño/Prototipo



Prueba de usabilidad:

1

➤ Objetivo

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

2

3

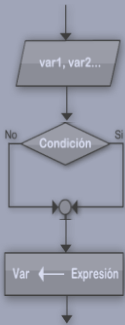
➤ Metodología

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

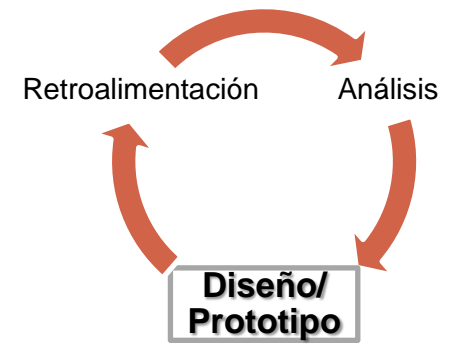
4

➤ Resultados

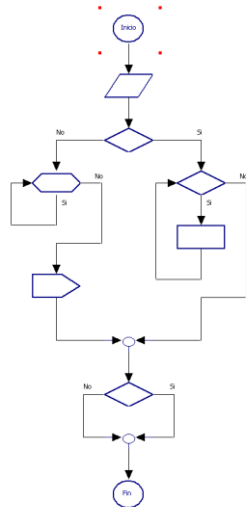
Retroalimentación interfaz.



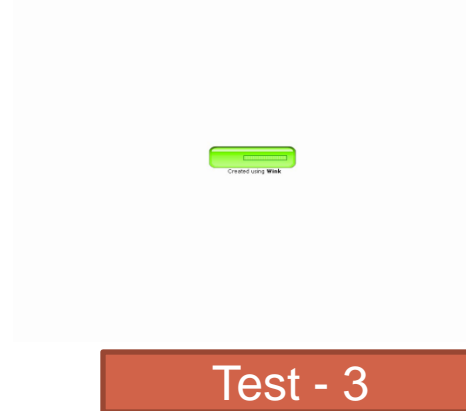
Actividad - Estudiantes



1) Desarrollar un diagrama



2) Agregar instrucciones



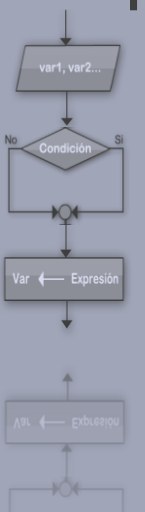
3) Eliminar un símbolo

1

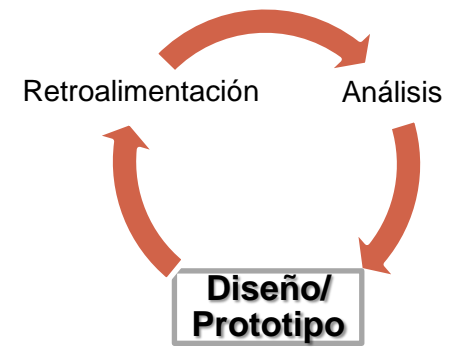
2

3

4



Actividad - Estudiantes



Contestar las siguientes preguntas:

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

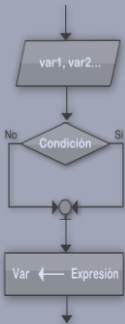
1

2

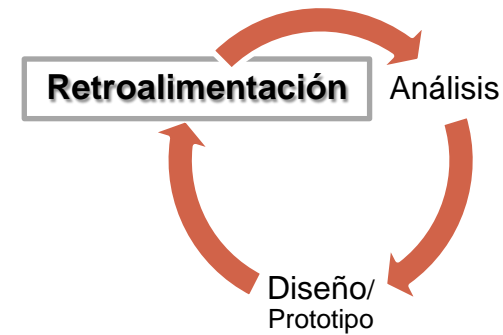
3

4

- ¿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

3

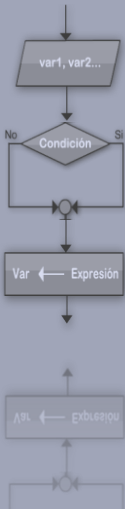
➤ Mejorar accesibilidad a las funciones

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

4

➤ Desarrollar nuevas funcionalidades

- Paso a paso
- Exportar código C/C++



Características de Origami

1

- Accesos intuitivos

2

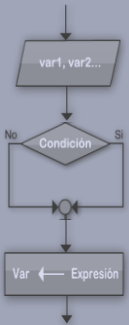
- Ejecución del diagrama de flujo

3

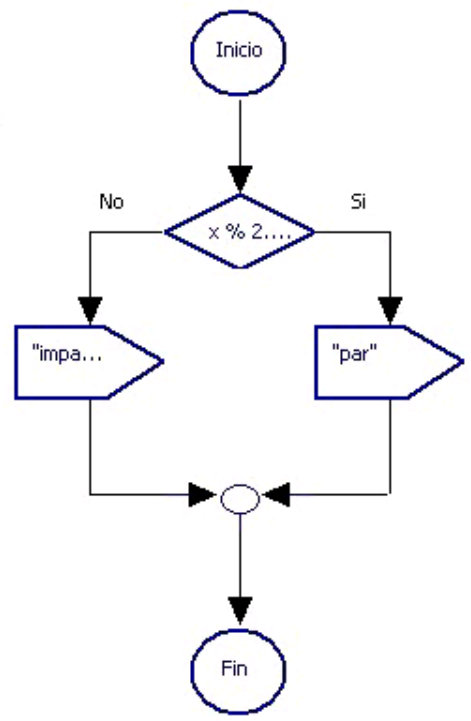
- Generar código C/C++

4

- Almacena información sobre la construcción del diagrama de flujo



- 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:

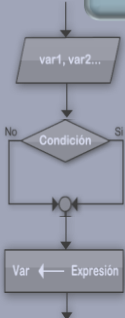
Carga Académica, uso de herramientas de programación, trabajo colegiado de profesores

1

2

3

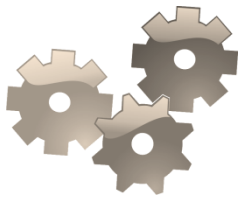
4



Resultados Iniciales



Archivo .Org



- ❖ Tiempo total que estuvo abierto el archivo
- ❖ Bloques utilizados en el diagrama
- ❖ Errores de compilación
- ❖ Aspectos comunes que dificultan una tarea en específico.
- ❖ Análisis de avance individual o grupal

```
Inicio nuevo diagrama
```

```
Inicio  
Fin
```

```
Al Se agregó una nueva figura al diagrama  
La figura agregada es de tipo "si".
```

```
Inicio
```

```
Entrada (Leer: int x)
```

```
If(x%2==0) {
```

```
  Si
```

```
}
```

```
Fin
```

```
/Ec - Error en la compilación: <ERROR>
```

```
main.cpp: In function 'int main (int char*)'
```

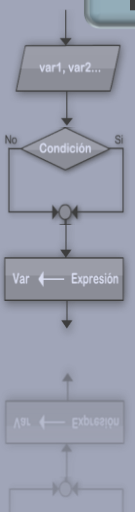
```
main.cpp:7: error: 'i' undeclared (first use this function)
```

```
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
```

1
2
3
4



Trabajo Futuro

1

- Refactorización
 - Código entendible
 - Fácil de mantener

2

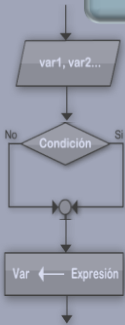
- Desarrollo de herramienta de análisis de la construcción de DF

3

- Analizar el avance de los estudiantes y modelo lógico-computacional

4

- 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

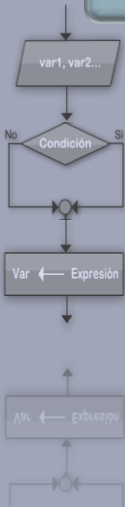
- Dinámica en la clase
- Almacenar información

1

2

3

4



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

Juan Ku Quintana

juankuq@gmail.com

Víctor Rodríguez Cámara

victor.fmat@gmail.com

Edgar Cambranes Martínez

edgar.cambranes@uady.mx



UADY

FACULTAD DE
MATEMÁTICAS

"Luz, Ciencia y Verdad"

„Лгш' Сгелсгс ь Vелгдсд„



UNIVERSIDAD
AUTÓNOMA DE