

## Paradigmas Fundamentales de Programación

### Introducción al modelo concurrente declarativo

Juan Francisco Díaz Frias

Maestría en Ingeniería, Énfasis en Ingeniería de Sistemas y Computación  
Escuela de Ingeniería de Sistemas y Computación,  
home page: <http://eisc.univalle.edu.co>  
Universidad del Valle - Cali, Colombia

## Plan

- 1 Generalidades
  - Concurrencia y concurrencia declarativa
  - No determinismo no observable
- 2 Desarrollo de este paradigma

## Plan

- 1 Generalidades
  - Concurrencia y concurrencia declarativa
  - No determinismo no observable
- 2 Desarrollo de este paradigma

## Plan

- 1 Generalidades
  - Concurrencia y concurrencia declarativa
  - No determinismo no observable
- 2 Desarrollo de este paradigma

## Generalidades

### Concurrencia

- Algunos programas se escriben mejor como un conjunto de actividades que se ejecutan independientemente. Tales programas son llamados **concurrentes**.
- La concurrencia es esencial en programas que interactúan con su ambiente, e.g., para agentes, programación de interfaces gráficas de usuario (GUI), interacción de sistemas operativos (OS), ...
- La concurrencia también permite organizar los programas en partes que se ejecutan independientemente e interactúan sólo cuando lo necesitan, i.e., programas cliente/servidor y productor/consumidor.

### Concurrencia declarativa

## Generalidades

### Concurrencia

- Algunos programas se escriben mejor como un conjunto de actividades que se ejecutan independientemente. Tales programas son llamados **concurrentes**.
- La concurrencia es esencial en programas que interactúan con su ambiente, e.g., para agentes, programación de interfaces gráficas de usuario (GUI), interacción de sistemas operativos (OS), ...
- La concurrencia también permite organizar los programas en partes que se ejecutan independientemente e interactúan sólo cuando lo necesitan, i.e., programas cliente/servidor y productor/consumidor.

### Concurrencia declarativa

- Enfoque alternativo declarativo al paradigma concurrente secuencialmente declarativo.

En este caso, los programas se describen como conjuntos de reglas que se aplican a los datos.

## Generalidades

### Concurrencia

- Algunos programas se escriben mejor como un conjunto de actividades que se ejecutan independientemente. Tales programas son llamados **concurrentes**.
- La concurrencia es esencial en programas que interactúan con su ambiente, e.g., para agentes, programación de interfaces gráficas de usuario (GUI), interacción de sistemas operativos (OS), ...
- La concurrencia también permite organizar los programas en partes que se ejecutan independientemente e interactúan sólo cuando lo necesitan, i.e., programas cliente/servidor y productor/consumidor.

### Concurrencia declarativa

- Extiende el modelo declarativo añadiéndole concurrencia permaneciendo declarativo.
  - Todas las técnicas de programación de programación para programas declarativos siguen siendo válidas.
- La concurrencia declarativa es una extensión del modelo de programación declarativa que permite la ejecución concurrente de programas declarativos.

## Generalidades

### Concurrencia

- Algunos programas se escriben mejor como un conjunto de actividades que se ejecutan independientemente. Tales programas son llamados **concurrentes**.
- La concurrencia es esencial en programas que interactúan con su ambiente, e.g., para agentes, programación de interfaces gráficas de usuario (GUI), interacción de sistemas operativos (OS), ...
- La concurrencia también permite organizar los programas en partes que se ejecutan independientemente e interactúan sólo cuando lo necesitan, i.e., programas cliente/servidor y productor/consumidor.

### Concurrencia declarativa

- Extiende el modelo declarativo añadiéndole concurrencia permaneciendo declarativo.
- Todas las técnicas de programación de razonamiento para programas declarativos siguen siendo válidas.
- La intuición subyacente está basada en el hecho que una variable de flujo de datos puede ser ligada a un único valor.
- Consecuencias:



## Generalidades

### Concurrencia

- Algunos programas se escriben mejor como un conjunto de actividades que se ejecutan independientemente. Tales programas son llamados **concurrentes**.
- La concurrencia es esencial en programas que interactúan con su ambiente, e.g., para agentes, programación de interfaces gráficas de usuario (GUI), interacción de sistemas operativos (OS), ...
- La concurrencia también permite organizar los programas en partes que se ejecutan independientemente e interactúan sólo cuando lo necesitan, i.e., programas cliente/servidor y productor/consumidor.

### Concurrencia declarativa

- Extiende el modelo declarativo añadiéndole concurrencia permaneciendo declarativo.
- Todas las técnicas de programación de razonamiento para programas declarativos siguen siendo válidas.
- La intuición subyacente está basada en el hecho que una variable de flujo de datos puede ser ligada a un único valor.
- Consecuencias:

## Generalidades

### Concurrencia

- Algunos programas se escriben mejor como un conjunto de actividades que se ejecutan independientemente. Tales programas son llamados **concurrentes**.
- La concurrencia es esencial en programas que interactúan con su ambiente, e.g., para agentes, programación de interfaces gráficas de usuario (GUI), interacción de sistemas operativos (OS), ...
- La concurrencia también permite organizar los programas en partes que se ejecutan independientemente e interactúan sólo cuando lo necesitan, i.e., programas cliente/servidor y productor/consumidor.

### Concurrencia declarativa

- Extiende el modelo declarativo añadiéndole concurrencia permaneciendo declarativo.
- Todas las técnicas de programación de razonamiento para programas declarativos siguen siendo válidas.
- La intuición subyacente está basada en el hecho que una variable de flujo de datos puede ser ligada a un único valor.
- Consecuencias:

El resultado de un programa es el mismo sea o no concurrente.

El programa puede ser escrito de forma más sencilla y concisa.

## Generalidades

### Concurrencia

- Algunos programas se escriben mejor como un conjunto de actividades que se ejecutan independientemente. Tales programas son llamados **concurrentes**.
- La concurrencia es esencial en programas que interactúan con su ambiente, e.g., para agentes, programación de interfaces gráficas de usuario (GUI), interacción de sistemas operativos (OS), ...
- La concurrencia también permite organizar los programas en partes que se ejecutan independientemente e interactúan sólo cuando lo necesitan, i.e., programas cliente/servidor y productor/consumidor.

### Concurrencia declarativa

- Extiende el modelo declarativo añadiéndole concurrencia permaneciendo declarativo.
- Todas las técnicas de programación de razonamiento para programas declarativos siguen siendo válidas.
- La intuición subyacente está basada en el hecho que una variable de flujo de datos puede ser ligada a un único valor.
- Consecuencias:
  - El resultado de un programa **es el mismo** sea o no concurrente.
  - El resultado de un programa se puede calcular **incrementalmente**.

## Generalidades

### Concurrencia

- Algunos programas se escriben mejor como un conjunto de actividades que se ejecutan independientemente. Tales programas son llamados **concurrentes**.
- La concurrencia es esencial en programas que interactúan con su ambiente, e.g., para agentes, programación de interfaces gráficas de usuario (GUI), interacción de sistemas operativos (OS), ...
- La concurrencia también permite organizar los programas en partes que se ejecutan independientemente e interactúan sólo cuando lo necesitan, i.e., programas cliente/servidor y productor/consumidor.

### Concurrencia declarativa

- Extiende el modelo declarativo añadiéndole concurrencia permaneciendo declarativo.
- Todas las técnicas de programación de razonamiento para programas declarativos siguen siendo válidas.
- La intuición subyacente está basada en el hecho que una variable de flujo de datos puede ser ligada a un único valor.
- Consecuencias:
  - El resultado de un programa **es el mismo** sea o no concurrente.
  - El resultado de un programa se puede calcular **incrementalmente**.

## Generalidades

### Concurrencia

- Algunos programas se escriben mejor como un conjunto de actividades que se ejecutan independientemente. Tales programas son llamados **concurrentes**.
- La concurrencia es esencial en programas que interactúan con su ambiente, e.g., para agentes, programación de interfaces gráficas de usuario (GUI), interacción de sistemas operativos (OS), ...
- La concurrencia también permite organizar los programas en partes que se ejecutan independientemente e interactúan sólo cuando lo necesitan, i.e., programas cliente/servidor y productor/consumidor.

### Concurrencia declarativa

- Extiende el modelo declarativo añadiéndole concurrencia permaneciendo declarativo.
- Todas las técnicas de programación de razonamiento para programas declarativos siguen siendo válidas.
- La intuición subyacente está basada en el hecho que una variable de flujo de datos puede ser ligada a un único valor.
- Consecuencias:
  - El resultado de un programa **es el mismo** sea o no concurrente.
  - El resultado de un programa se puede calcular **incrementalmente**.

## Plan

- 1 Generalidades
  - Concurrencia y concurrencia declarativa
  - No determinismo no observable
- 2 Desarrollo de este paradigma

## Generalidades (2)

Considere el problema de calcular una lista de cuadrados de enteros sucesivos.

### Versión declarativa

```

fun {Gen L H}
  {Delay 100}
  if L>H then nil
  else L|{Gen L+1 H} end
end
Xs={Gen 1 10}
Ys={Map Xs fun {$ X}
      X*X
    end}
{Browse Ys}
  
```

### Versión concurrente declarativa

```

thread Xs={Gen 1 10} end
thread Ys={Map Xs
      fun {$ X}
        X*X
      end}
end
{Browse Ys}
  
```

¿Cuál es la diferencia entre las dos versiones?

Gran ventaja: no-determinismo no observable.

## Generalidades (2)

Considere el problema de calcular una lista de cuadrados de enteros sucesivos.

### Versión declarativa

```

fun {Gen L H}
  {Delay 100}
  if L>H then nil
  else L|{Gen L+1 H} end
end
Xs={Gen 1 10}
Ys={Map Xs fun {$ X}
      X*X
    end}
{Browse Ys}
  
```

### Versión concurrente declarativa

```

thread Xs={Gen 1 10} end
thread Ys={Map Xs
      fun {$ X}
        X*X
      end}
end
{Browse Ys}
  
```

¿Cuál es la diferencia entre las dos versiones?

Gran ventaja: no-determinismo no observable.



## Generalidades (2)

Considere el problema de calcular una lista de cuadrados de enteros sucesivos.

### Versión declarativa

```

fun {Gen L H}
  {Delay 100}
  if L>H then nil
  else L|{Gen L+1 H} end
end
Xs={Gen 1 10}
Ys={Map Xs fun {$ X}
      X*X
    end}
{Browse Ys}
  
```

### Versión concurrente declarativa

```

thread Xs={Gen 1 10} end
thread Ys={Map Xs
      fun {$ X}
        X*X
      end}
end
{Browse Ys}
  
```

¿Cuál es la diferencia entre las dos versiones?

Gran ventaja: **no-determinismo no observable.**

## Desarrollo de este paradigma

- 1 Concurrencia dirigida por los datos: el concepto de hilo
- 2 El concepto de flujo
- 3 Concurrencia dirigida por la demanda: el concepto de disparador por necesidad
- 4 Evaluación perezosa
- 5 Limitaciones y extensiones de la programación declarativa

## Desarrollo de este paradigma

- 1 Concurrencia dirigida por los datos: el concepto de hilo
- 2 El concepto de flujo
- 3 Concurrencia dirigida por la demanda: el concepto de disparador por necesidad
- 4 Evaluación perezosa
- 5 Limitaciones y extensiones de la programación declarativa

## Desarrollo de este paradigma

- 1 Concurrencia dirigida por los datos: el concepto de hilo
- 2 El concepto de flujo
- 3 Concurrencia dirigida por la demanda: el concepto de disparador por necesidad
- 4 Evaluación perezosa
- 5 Limitaciones y extensiones de la programación declarativa

## Desarrollo de este paradigma

- 1 Concurrencia dirigida por los datos: el concepto de hilo
- 2 El concepto de flujo
- 3 Concurrencia dirigida por la demanda: el concepto de disparador por necesidad
- 4 Evaluación perezosa
- 5 Limitaciones y extensiones de la programación declarativa

## Desarrollo de este paradigma

- 1 Concurrencia dirigida por los datos: el concepto de hilo
- 2 El concepto de flujo
- 3 Concurrencia dirigida por la demanda: el concepto de disparador por necesidad
- 4 Evaluación perezosa
- 5 Limitaciones y extensiones de la programación declarativa