

Bienvenida + Introducción a Gobstones

Francisco Soullignac

Tecnicatura en programación informática,
Licenciatura en Desarrollo de Software,
Universidad Nacional de Quilmes

Introducción a la programación

¿Qué es programar?

Introducción a Gobstones (¿qué es un programa?)

La física de Gobstones

Programas

Documentación

Precondición

Documentación

Resumen

¿Qué es programar?

¿Qué es programar?

Comunicar cómo se resuelve un problema de forma metódica o mecánica

¿Qué es programar?

Comunicar cómo se resuelve un **problema** de forma metódica o mecánica

- ▶ **Problema**: hacer algo con ciertas **condiciones iniciales**

¿Qué es programar?

Comunicar cómo se resuelve un problema de forma metódica o mecánica

- ▶ Problema: hacer algo con ciertas condiciones iniciales
- ▶ El mensaje se conoce como **programa** y se puede escribir en algún **lenguaje** (castellano, por ejemplo)

¿Qué es programar?

Comunicar cómo se resuelve un problema de forma **metódica o mecánica**

- ▶ Problema: hacer algo con ciertas condiciones iniciales
- ▶ El mensaje se conoce como programa y se puede escribir en algún lenguaje (castellano, por ejemplo)
- ▶ **Metódico** significa que el programa **siempre** se ejecuta de la misma forma (y no tiene ambigüedades)

¿Qué es programar?

Comunicar cómo se resuelve un problema de forma metódica o mecánica

- ▶ Problema: hacer algo con ciertas condiciones iniciales
- ▶ El mensaje se conoce como programa y se puede escribir en algún lenguaje (castellano, por ejemplo)
- ▶ Metódico significa que el programa **siempre** se ejecuta de la misma forma (y no tiene ambigüedades)
- ▶ Receptores del programa:
 - ▶ Ejecutor/es: aquel que obedece el programa sin discusión
 - ▶ Revisor/es: deben **comprender** el programa y pueden modificarlo

¿Qué es programar?

Comunicar cómo se resuelve un problema de forma metódica o mecánica

- ▶ Problema: hacer algo con ciertas condiciones iniciales
- ▶ El mensaje se conoce como programa y se puede escribir en algún lenguaje (castellano, por ejemplo)
- ▶ Metódico significa que el programa **siempre** se ejecuta de la misma forma (y no tiene ambigüedades)
- ▶ Receptores del programa:
 - ▶ Ejecutor/es: aquel que obedece el programa sin discusión
 - ▶ Revisor/es: deben comprender el programa y pueden modificarlo

Es importante que el programa sea **entendible**

¿Qué es la programación informática?

¿Qué es la programación informática?

Comunicar cómo se resuelve un problema en una computadora

¿Qué es la programación informática?

Comunicar cómo se resuelve un problema en una computadora

- ▶ Receptores del programa:
 - ▶ Computadora: que ejecuta lo que dice el programa
 - ▶ Programadores: que leen el programa y pueden modificarlo
 - ▶ La computadora siempre entiende; el programador no

¿Qué es la programación informática?

Comunicar cómo se resuelve un problema en una computadora

- ▶ Receptores del programa:
 - ▶ Computadora: que ejecuta lo que dice el programa
 - ▶ Programadores: que leen el programa y pueden modificarlo
 - ▶ La computadora siempre entiende; el programador no
- ▶ Necesitamos un lenguaje de **programación**
 - ▶ Que describe ideas ejecutables por una computadora
 - ▶ Que sea agradable de leer por humanos

¿Qué es la programación informática?

Comunicar cómo se resuelve un problema en una computadora

- ▶ Receptores del programa:
 - ▶ Computadora: que ejecuta lo que dice el programa
 - ▶ Programadores: que leen el programa y pueden modificarlo
 - ▶ La computadora siempre entiende; el programador no
- ▶ Necesitamos un lenguaje de programación
 - ▶ Que describe ideas ejecutables por una computadora
 - ▶ Que sea agradable de leer por humanos
- ▶ El lenguaje no hace al virtuoso
 - ▶ Se puede escribir mal en cualquier lenguaje
 - ▶ Se puede ganar un nobel en cualquier lenguaje

¿Qué es la programación informática?

Comunicar cómo se resuelve un problema en una computadora

- ▶ Receptores del programa:
 - ▶ Computadora: que ejecuta lo que dice el programa
 - ▶ Programadores: que leen el programa y pueden modificarlo
 - ▶ La computadora siempre entiende; el programador no
- ▶ Necesitamos un lenguaje de programación
 - ▶ Que describe ideas ejecutables por una computadora
 - ▶ Que sea agradable de leer por humanos
- ▶ El lenguaje no hace al virtuoso
 - ▶ Se puede escribir mal en cualquier lenguaje
 - ▶ Se puede ganar un nobel en cualquier lenguaje

La computadora siempre entiende; el programador no

Elementos de un lenguaje

Elementos a respetar

- ▶ Símbolos o letras (a, 2, &, ϕ , æ)

Elementos de un lenguaje

Elementos a respetar

- ▶ Símbolos o letras (a, 2, &, ϕ , æ)
- ▶ Palabras (pepe, kkvw)

Elementos de un lenguaje

Elementos a respetar

- ▶ Símbolos o letras (a, 2, &, ϕ , æ)
- ▶ Palabras (pepe, kkvzw)
- ▶ Reglas de sintaxis
 - ▶ Voy a la facultad por la tarde
 - ▶ facultad tarde a por voy la la

Elementos de un lenguaje

Elementos a respetar

- ▶ Símbolos o letras (a, 2, &, ϕ , æ)
- ▶ Palabras (pepe, kkvw)
- ▶ Reglas de sintaxis
 - ▶ Voy a la facultad por la tarde
 - ▶ facultad tarde a por voy la la
- ▶ Semántica
 - ▶ Me río en el baño
 - ▶ Me baño en el río

Elementos de un lenguaje

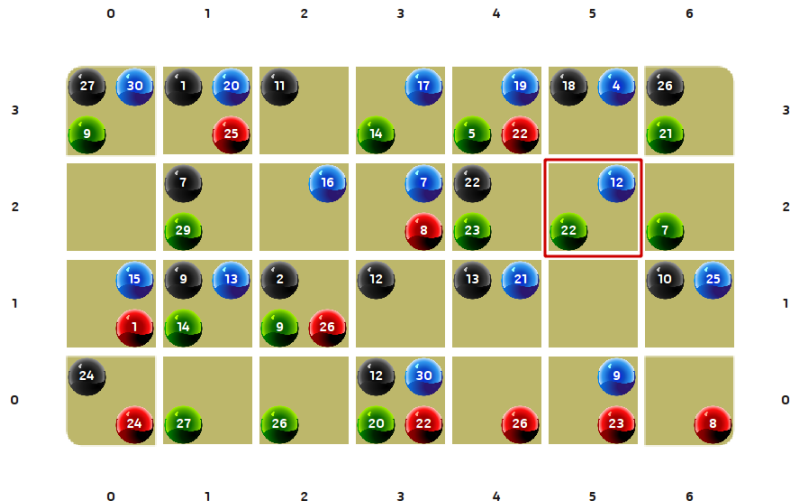
Elementos a respetar

- ▶ Símbolos o letras (a, 2, &, ϕ , æ)
- ▶ Palabras (pepe, kkvw)
- ▶ Reglas de sintaxis
 - ▶ Voy a la facultad por la tarde
 - ▶ facultad tarde a por voy la la
- ▶ Semántica
 - ▶ Me río en el baño
 - ▶ Me baño en el río

Elementos propios de cada persona

- ▶ Estilo
 - ▶ Indentación (sangrado)
 - ▶ Interlineado
 - ▶ Saltos de página, etc.

Empezando con Gobstones



Elementos de Gobstones I: bolitas

Bolitas

- ▶ Cuerpo tridimensional en que cada punto equidista del centro

Elementos de Gobstones I: bolitas

Bolitas

- ▶ Canica :)

Elementos de Gobstones I: bolitas

Bolitas

- ▶ Canica :)
- ▶ De colores **Azul**, Negro, **Rojo** y **Verde**



Bolitas: Las dibujamos en dos dimensiones

Elementos de Gobstones II: tablero

Tablero

- ▶ Rectangular y dividido en una cantidad **finita** de **celdas**
- ▶ Cada celda tiene una cantidad **finita** de bolitas
 - ▶ Ojo: finito no significa acotada



Elementos de Gobstones III: cabezal

Cabezal

- Siempre posicionado encima de la **celda actual** del tablero



Elementos de Gobstones III: cabezal

Cabezal

- ▶ Siempre posicionado encima de la **celda actual** del tablero
- ▶ Puede insertar o sacar bolitas de la **celda actual**



Elementos de Gobstones III: cabezal

Cabezal

- ▶ Siempre posicionado encima de la **celda actual** del tablero
- ▶ Puede insertar o sacar bolitas de la **celda actual**
- ▶ Se puede mover en forma **horizontal** y **vertical** por el tablero



Qué es un programa Gobstones

- ▶ Máquina especial que modela una computadora
 - ▶ Estado: combinación de tablero + bolitas + cabezal

Qué es un programa Gobstones

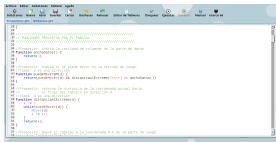
- ▶ Máquina especial que modela una computadora
 - ▶ Estado: combinación de tablero + bolitas + cabezal
- ▶ Programa: **Texto** que describe cómo transformar un estado inicial en un estado final (*)

[illegible]

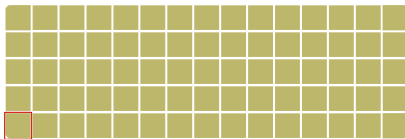
programa

Qué es un programa Gobstones

- ▶ Máquina especial que modela una computadora
 - ▶ Estado: combinación de tablero + bolitas + cabezal
- ▶ Programa: Texto que describe cómo transformar un estado inicial en un estado final (*)



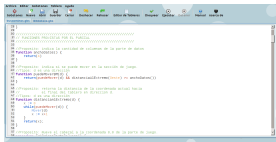
programa



estado inicial

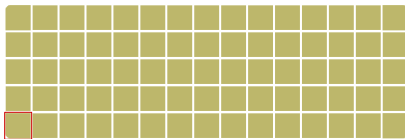
Qué es un programa Gobstones

- ▶ Máquina especial que modela una computadora
 - ▶ Estado: combinación de tablero + bolitas + cabezal
- ▶ Programa: Texto que describe cómo transformar un estado inicial en un estado **final** (*)

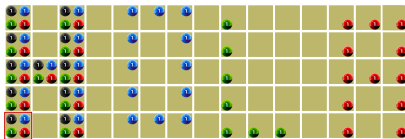


```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

programa



estado inicial



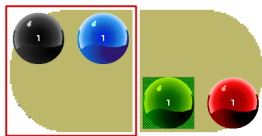
estado final

Un ejemplo

```
1 program
2 {
3   Poner( Verde )
4   Poner( Rojo )
5   Mover( Este )
6   Poner( Negro )
7   Poner( Azul )
8   Mover( Oeste )
9 }
```

Un ejemplo

```
1 program
2 {
3   Poner( Verde )
4   Poner( Rojo )
5   Mover( Este )
6   Poner( Negro )
7   Poner( Azul )
8   Mover( Oeste )
9 }
```



estado inicial

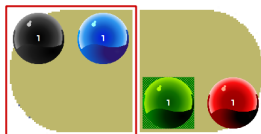
estado final

Un ejemplo

```

1  program
2  {
3      Poner ( Verde )
4      Poner ( Rojo )
5      Mover ( Este )
6      Poner ( Negro )
7      Poner ( Azul )
8      Mover ( Oeste )
9  }

```



estado inicial



estado final

Cómo escribo un programa I: valores y comandos

Denotación: dentro del programa puedo referirme a

- ▶ Colores: Azul, Negro, Rojo, Verde
- ▶ Direcciones: Norte, Este, Sur, Oeste

Cómo escribo un programa I: valores y comandos

Denotación: dentro del programa puedo referirme a

- ▶ Colores: Azul, Negro, Rojo, Verde
- ▶ Direcciones: Norte, Este, Sur, Oeste

Comandos: instruyen al cabezal

Cómo escribo un programa I: valores y comandos

Denotación: dentro del programa puedo referirme a

- ▶ Colores: Azul, Negro, Rojo, Verde
- ▶ Direcciones: Norte, Este, Sur, Oeste

Comandos: instruyen al cabezal

- ▶ Si c denota un color y d denota una dirección

Cómo escribo un programa I: valores y comandos

Denotación: dentro del programa puedo referirme a

- ▶ Colores: Azul, Negro, Rojo, Verde
- ▶ Direcciones: Norte, Este, Sur, Oeste

Comandos: instruyen al cabezal

- ▶ Si c denota un color y d denota una dirección
- ▶ **Poner**(c): pone una bolita de color c en la celda actual

Cómo escribo un programa I: valores y comandos

Denotación: dentro del programa puedo referirme a

- ▶ Colores: Azul, Negro, Rojo, Verde
- ▶ Direcciones: Norte, Este, Sur, Oeste

Comandos: instruyen al cabezal

- ▶ Si c denota un color y d denota una dirección
- ▶ $\text{Poner}(c)$: pone una bolita de color c en la celda actual
- ▶ $\text{Sacar}(c)$: saca una bolita de color c de la celda actual

Cómo escribo un programa I: valores y comandos

Denotación: dentro del programa puedo referirme a

- ▶ Colores: Azul, Negro, Rojo, Verde
- ▶ Direcciones: Norte, Este, Sur, Oeste

Comandos: instruyen al cabezal

- ▶ Si c denota un color y d denota una dirección
- ▶ $\text{Poner}(c)$: pone una bolita de color c en la celda actual
- ▶ $\text{Sacar}(c)$: saca una bolita de color c de la celda actual
- ▶ **Mover**(d): mueve el cabezal a la celda lindante en dirección d

Cómo escribo un programa I: valores y comandos

Denotación: dentro del programa puedo referirme a

- ▶ Colores: Azul, Negro, Rojo, Verde
- ▶ Direcciones: Norte, Este, Sur, Oeste

Comandos: instruyen al cabezal

- ▶ Si c denota un color y d denota una dirección
- ▶ Poner(c): pone una bolita de color c en la celda actual
- ▶ Sacar(c): saca una bolita de color c de la celda actual
- ▶ Mover(d): mueve el cabezal a la celda lindante en dirección d
- ▶ Ejemplos:
 - ▶ Poner(Rojo), Sacar(Verde), Mover(Este)

Más reglas de sintaxis y semántica

- ▶ Los colores y las direcciones se escriben en mayúscula
 - ▶ azul, AZUL, AZul, azUL, etc no son palabras en Gobstones

Más reglas de sintaxis y semántica

- ▶ Los colores y las direcciones se escriben en mayúscula
 - ▶ azul, AZUL, AZul, azUL, etc no son palabras en Gobstones
- ▶ No confundir los colores del lenguaje Gobstones con el castellano
 - ▶ Amarillo no es una palabra en Gobstones

Más reglas de sintaxis y semántica

- ▶ Los colores y las direcciones se escriben en mayúscula
 - ▶ azul, AZUL, AZul, azUL, etc no son palabras en Gobstones
- ▶ No confundir los colores del lenguaje Gobstones con el castellano
 - ▶ Amarillo no es una palabra en Gobstones
- ▶ Los comandos se escriben en mayúscula
 - ▶ poner y PONER no son comandos en Gobstones

Más reglas de sintaxis y semántica

- ▶ Los colores y las direcciones se escriben en mayúscula
 - ▶ azul, AZUL, AZul, azUL, etc no son palabras en Gobstones
- ▶ No confundir los colores del lenguaje Gobstones con el castellano
 - ▶ Amarillo no es una palabra en Gobstones
- ▶ Los comandos se escriben en mayúscula
 - ▶ poner y PONER no son comandos en Gobstones
- ▶ Poner, Mover y Sacar requieren **argumentos** para funcionar
 - ▶ Es lo que escribimos entre paréntesis
 - ▶ Los argumentos de Poner y Sacar deben denotar colores
 - ▶ El argumento de Mover debe denotar una dirección

Cómo escribo un programa II: secuencia

- ▶ Los comandos se pueden secuenciar uno detrás de otro
 - ▶ Separándolos con fin de línea o punto y coma

```
1 Poner( Verde ); Mover( Este )  
2 Poner( Rojo )  
3 Mover( Sur );
```

Cómo escribo un programa II: secuencia

- ▶ Los comandos se pueden secuenciar uno detrás de otro
 - ▶ Separándolos con fin de línea o punto y coma

```
1 Poner( Verde ); Mover( Este )  
2 Poner( Rojo )  
3 Mover( Sur );
```

- ▶ La secuencia se **ejecuta** de acuerdo al **orden de lectura**
 - ▶ El orden en que se escriben los comandos es importante

Cómo escribo un programa II: secuencia

- ▶ Los comandos se pueden secuenciar uno detrás de otro
 - ▶ Separándolos con fin de línea o punto y coma

Cómo escribo un programa III: bloque

- ▶ Una secuencia de comandos se puede agrupar en un **bloque**
 - ▶ Encerrando la secuencia entre llaves { y }
 - ▶ Se usan para estructurar código (*)

```
1 {  
2   _ _ Poner ( Azul )  
3   _ _ {  
4       _ _ _ _ Poner ( Verde ); _ _ Poner ( Verde )  
5       _ _ _ _ Poner ( Rojo )  
6   _ _ }  
7   _ _ Poner ( Negro )  
8 }
```


Cómo escribo un programa III: bloque

- ▶ Una secuencia de comandos se puede agrupar en un **bloque**
 - ▶ Encerrando la secuencia entre llaves { y }
 - ▶ Se usan para estructurar código (*)
- ▶ Los bloques pueden **anidarse**

```
1 {  
2   _ _ Poner ( Azul )  
3   _ _ {  
4       _ _ _ _ Poner ( Verde ); _ _ Poner ( Verde )  
5       _ _ _ _ Poner ( Rojo )  
6   _ _ }  
7   _ _ Poner ( Negro )  
8 }
```

El bloque rojo está **anidado** dentro del verde

Cómo escribo un programa III: bloque

- ▶ Una secuencia de comandos se puede agrupar en un **bloque**
 - ▶ Encerrando la secuencia entre llaves { y }
 - ▶ Se usan para estructurar código (*)
- ▶ Los bloques pueden **anidarse**
- ▶ Tener en cuenta la **indentación**

```
1 {  
2   _ _ Poner ( Azul )  
3   _ _ {  
4       _ _ _ _ Poner ( Verde ); _ _ Poner ( Verde )  
5       _ _ _ _ Poner ( Rojo )  
6   _ _ }  
7   _ _ Poner ( Negro )  
8 }
```

Cómo escribo un programa IV: program

- ▶ Palabra reservada: **program**

Cómo escribo un programa IV: program

- ▶ Palabra reservada: **program**
- ▶ Se escribe `program` `< bloque >`

Cómo escribo un programa IV: program

- ▶ Palabra reservada: **program**
- ▶ Se escribe `program` `{` bloque `}`
- ▶ Dentro del texto hay un único `program`
 - ▶ Gobstones ejecuta el bloque de `program`

Cómo escribo un programa IV: program

- ▶ Palabra reservada: **program**
- ▶ Se escribe `program` `{` bloque `}`
- ▶ Dentro del texto hay un único `program`
 - ▶ Gobstones ejecuta el bloque de `program`

```
1 program
2 {
3   Poner( Verde )
4   Poner( Rojo )
5   Mover( Este )
6   Poner( Negro )
7   Poner( Azul )
8   Mover( Oeste )
9 }
```

Cómo escribo un programa V: comentarios

- ▶ En Gobstones es posible escribir texto a ser ignorado
 - ▶ Para incluir **comentarios** en el código
 - ▶ Para documentar el código
 - ▶ En fin, para la comunicación con humanos

Cómo escribo un programa V: comentarios

- ▶ En Gobstones es posible escribir texto a ser ignorado
 - ▶ Para incluir **comentarios** en el código
 - ▶ Para documentar el código
 - ▶ En fin, para la comunicación con humanos
- ▶ Dos sabores
 - ▶ En una línea: `//línea a ignorar`
 - ▶ En varias líneas: `/*texto a ignorar*/`

```
1  /* Proposito: pone una bolita de cada color
2   * Cabecal: queda en la celda inicial
3   */
4  program {
5     //Recordar: colores Azul, Negro, Rojo, Verde
6     Poner(Azul)
7     Poner(Negro)
8     Poner(Rojo)
9     Poner(Verde)
10 }
```


Ejercitando

- ▶ Escribir un programa que ponga dos bolitas verdes en la **celda inicial**

Ejercitando

- ▶ Escribir un programa que ponga dos bolitas verdes en la **celda inicial**
- ▶ Escribir un programa que ponga una bolita verde al este de la **celda inicial** y otra azul al oeste de la **celda inicial**

Ejercitando

- ▶ Escribir un programa que ponga dos bolitas verdes en la **celda inicial**
- ▶ Escribir un programa que ponga una bolita verde al este de la **celda inicial** y otra azul al oeste de la **celda inicial**
- ▶ Escribir un programa que ponga una bolita de cada color en la **celda inicial**.

Ejercitando

- ▶ Escribir un programa que ponga dos bolitas verdes en la **celda inicial**
- ▶ Escribir un programa que ponga una bolita verde al este de la **celda inicial** y otra azul al oeste de la **celda inicial**
- ▶ Escribir un programa que ponga una bolita de cada color en la **celda inicial**.
- ▶ Escribir un programa que ponga una bolita de color azul en la celda lindante al Norte de la actual, dejando el cabezal en la **celda inicial**.

- ▶ ¿Documentamos bien nuestro programa?
- ▶ ¿Dónde queda el cabezal en cada programa?

Efecto de un programa

- ▶ Qué transformación se realiza al tablero y cabezal
 - ▶ No importan los efectos parciales; sólo importa el efecto global

Efecto de un programa

- ▶ Qué transformación se realiza al tablero y cabezal
 - ▶ No importan los efectos parciales; sólo importa el efecto global
- ▶ Programas **equivalentes**: cuando producen el mismo efecto
 - ▶ Independientemente de cuál sea el estado inicial

```
1
2 program {
3   Poner ( Verde )
4   Poner ( Rojo )
5 }
```

```
1
2 program {
3   Poner ( Rojo )
4   Poner ( Verde )
5 }
```

Dos programas con el mismo efecto

Efecto de un programa

- ▶ Qué transformación se realiza al tablero y cabezal
 - ▶ No importan los efectos parciales; sólo importa el efecto global
- ▶ Programas **equivalentes**: cuando producen el mismo efecto
 - ▶ Independientemente de cuál sea el estado inicial

```
1 //Pone una bolita Roja y otra Verde
2 program {
3   Poner( Verde )
4   Poner( Rojo )
5 }
```

```
1 //Pone una bolita Roja y otra Verde
2 program {
3   Poner( Rojo )
4   Poner( Verde )
5 }
```

Dos programas con el mismo efecto

- ▶ Hay que documentar el efecto esperado en un comentario

Efecto de un programa

```
1  /* Pone una bolita Verde, luego una bolita Roja,  
2  * despues una Azul y finalmente una Negra  
3  */  
4  program {  
5      Poner( Verde )  
6      Poner( Rojo )  
7      Poner( Azul )  
8      Poner( Negro )  
9  }
```

- La documentación esta **mal**. ¿ porqué?

Efecto de un programa

```
1  /* Pone una bolita Verde, luego una bolita Roja,  
2  * despues una Azul y finalmente una Negra  
3  */  
4  program {  
5      Poner( Verde )  
6      Poner( Rojo )  
7      Poner( Azul )  
8      Poner( Negro )  
9  }
```

- ▶ La documentación esta **mal**. ¿ porqué?
- ▶ Notar que no importa cómo se llega al efecto

Propósito de un programa

- ▶ Problema Gobstones: transformar un tablero en otro

Propósito de un programa

- ▶ Problema Gobstones: transformar un tablero en otro
- ▶ Propósito: descripción del problema a resolver
 - ▶ Debería coincidir con el efecto, pero...
 - ▶ ...el programa podría tener errores...
 - ▶ ...que ocurren cuando el efecto no es el esperado

Propósito de un programa

- ▶ Problema Gobstones: transformar un tablero en otro
- ▶ Propósito: descripción del problema a resolver
 - ▶ Debería coincidir con el efecto, pero...
 - ▶ ...el programa podría tener errores...
 - ▶ ...que ocurren cuando el efecto no es el esperado
- ▶ El propósito se escribe junto al código del programa

Propósito de un programa

- ▶ Problema Gobstones: transformar un tablero en otro
- ▶ Propósito: descripción del problema a resolver
 - ▶ Debería coincidir con el efecto, pero...
 - ▶ ...el programa podría tener errores...
 - ▶ ...que ocurren cuando el efecto no es el esperado
- ▶ El propósito se escribe junto al código del programa
- ▶ No olvidarse el cabezal
 - ▶ Si no se indica, es que queda en un lugar no especificado.

Propósito de un programa

- ▶ Problema Gobstones: transformar un tablero en otro
- ▶ Propósito: descripción del problema a resolver
 - ▶ Debería coincidir con el efecto, pero...
 - ▶ ...el programa podría tener errores...
 - ▶ ...que ocurren cuando el efecto no es el esperado
- ▶ El propósito se escribe junto al código del programa
- ▶ No olvidarse el cabezal
 - ▶ Si no se indica, es que queda en un lugar no especificado.

```
1  /* Proposito: pone una bolita de cada color
2   * Cabezal: queda en la celda inicial
3   */
4  program {
5     Poner( Azul )
6     Poner( Negro )
7     Poner( Rojo )
8     Poner( Verde )
9  }
```

Este programa no cumple su propósito

Propósito de un programa

► Documente el siguiente programa

```
1  /* Proposito: ????
```

```
2  * Cabezal: ???
```

```
3  */
```

```
4  program {
```

```
5     Sacar(Azul)
```

```
6     Poner(Negro)
```

```
7 }
```

Correctitud: efecto vs. propósito

- ¿Cuál de los siguientes programas es *correcto*?

```
1  /* Pone dos bolitas azules en la celda actual
2   * Deja el cabezal en la celda actual
3   */
4  program {
5     Poner(Azul); Poner(Azul)
6  }
7
8  /* Pone dos bolitas azules en la celda actual
9   */
10 program {
11     Poner(Azul); Poner(Rojo)
12 }
13
14 program {
15     Poner(Rojo); Poner(Rojo)
16 }
```

- Un programa es **correcto** cuando su efecto **siempre** coincide con lo que indica su propósito (*).
- Sin propósito es **imposible** hablar de correctitud.

Precondiciones

- ¿Qué ocurre si ejecutamos el siguiente programa sobre un tablero con a lo sumo una bolita azul?

```
1  /* Proposito: Saca dos bolitas azules de la celda actual
2   * Cabezal: queda en la celda inicial
3   */
4  program {
5      Sacar(Azul)
6      Sacar(Azul)
7  }
```

Precondiciones

- ¿Qué ocurre si ejecutamos el siguiente programa sobre un tablero con a lo sumo una bolita azul?

```
1  /* Proposito: Saca dos bolitas azules de la celda actual
2   * Cabezal: queda en la celda inicial
3   */
4  program {
5     Sacar(Azul)
6     Sacar(Azul)
7  }
```

- El tablero no es **apropiado** para este programa...

Precondiciones

- ¿Qué ocurre si ejecutamos el siguiente programa sobre un tablero con a lo sumo una bolita azul?

```
1  /* Proposito: Saca dos bolitas azules de la celda actual
2   * Cabezal: queda en la celda inicial
3   */
4  program {
5     Sacar(Azul)
6     Sacar(Azul)
7  }
```

- El tablero no es **apropiado** para este programa...
- ... razón por la cual el programa no puede cumplir su propósito!

Precondiciones

- ¿Qué ocurre si ejecutamos el siguiente programa sobre un tablero con a lo sumo una bolita azul?

```
1  /* Proposito: Saca dos bolitas azules de la celda actual
2   * Cabezal: queda en la celda inicial
3   */
4  program {
5     Sacar(Azul)
6     Sacar(Azul)
7  }
```

- El tablero no es **apropiado** para este programa...
- ... razón por la cual el programa no puede cumplir su propósito!
- ¿Está mal el programa; está mal el propósito?

Precondiciones

- ▶ Precondición: condiciones necesarias para **garantizar** que el problema puede resolverse
 - ▶ Si se cumple, el programa **debe** cumplir su propósito
 - ▶ Si no, el programa es libre de tener **cualquier** efecto

Precondiciones

- ▶ Precondición: condiciones necesarias para **garantizar** que el problema puede resolverse
 - ▶ Si se cumple, el programa **debe** cumplir su propósito
 - ▶ Si no, el programa es libre de tener **cualquier** efecto
- ▶ La precondición se escribe junto al propósito

```
1  /* Proposito: Sacar dos bolitas azules de la celda actual
2   * Cabezal: queda en la celda inicial
3   * Precondicion: al menos dos bolitas azules en celda actual
4   */
5  program {
6      Sacar(Azul)
7      Sacar(Azul)
8  }
```

Precondiciones

- ▶ Precondición: condiciones necesarias para **garantizar** que el problema puede resolverse
 - ▶ Si se cumple, el programa **debe** cumplir su propósito
 - ▶ Si no, el programa es libre de tener **cualquier** efecto
- ▶ La precondición se escribe junto al propósito

```
1  /* Proposito: Sacar dos bolitas azules de la celda actual
2   * Cabecal: queda en la celda inicial
3   * Precondicion: al menos dos bolitas azules en celda actual
4   */
5  program {
6      Sacar(Azul)
7      Sacar(Azul)
8  }
```

- ▶ Un programa es **correcto** cuando su efecto coincide con lo que indica su propósito **siempre** que se cumple la precondición

Problema vs. programa (¿qué? vs ¿cómo?)

- ▶ Especificación de un problema: precondition + propósito
 - ▶ A este concepto también se lo llama contrato

Problema vs. programa (¿qué? vs ¿cómo?)

- ▶ Especificación de un problema: precondition + propósito
 - ▶ A este concepto también se lo llama contrato
- ▶ Dos problemas son **equivalentes** cuando tienen la misma especificación
 - ▶ Por ende, los programas que resuelven uno resuelven el otro

Problema vs. programa (¿qué? vs ¿cómo?)

- ▶ Especificación de un problema: precondition + propósito
 - ▶ A este concepto también se lo llama contrato
- ▶ Dos problemas son **equivalentes** cuando tienen la misma especificación
 - ▶ Por ende, los programas que resuelven uno resuelven el otro
- ▶ La especificación define el **qué**, el programa el **cómo**

Problema vs. programa (¿qué? vs ¿cómo?)

- ▶ Especificación de un problema: precondition + propósito
 - ▶ A este concepto también se lo llama contrato
- ▶ Dos problemas son **equivalentes** cuando tienen la misma especificación
 - ▶ Por ende, los programas que resuelven uno resuelven el otro
- ▶ La especificación define el **qué**, el programa el **cómo**
- ▶ Rol usuario: sólo importa el qué (especificación)
 - ▶ e.g., usar facebook, google, un avión, un remedio, etc

Problema vs. programa (¿qué? vs ¿cómo?)

- ▶ Especificación de un problema: precondition + propósito
 - ▶ A este concepto también se lo llama contrato
- ▶ Dos problemas son **equivalentes** cuando tienen la misma especificación
 - ▶ Por ende, los programas que resuelven uno resuelven el otro
- ▶ La especificación define el **qué**, el programa el **cómo**
- ▶ Rol usuario: sólo importa el qué (especificación)
 - ▶ e.g., usar facebook, google, un avión, un remedio, etc
- ▶ Rol programador: le importa tanto el qué como el cómo

Regla de la precondition débil

- ▶ Cuál es una precondition razonable para el problema de:
 1. sacar una bolita de cada color
 2. poner tres bolitas azules
 3. dibujar un cuadrado de 3×3 celdas centradas en la celda actual
 4. posicionar el cabezal sobre una celda con bolitas rojas
- ▶ La precondition debería ser lo **más débil** posible
 - ▶ De esta forma, se puede resolver más problemas

Regla de la precondition débil, ejemplo

- ▶ ¿Cuál debería ser la precondition de un programa que pone una línea de longitud 3 hacia el este con una bolita de color azul en cada celda, empezando en la celda actual?

Regla de la precondition débil, ejemplo

- ▶ ¿Cuál debería ser la precondition de un programa que pone una línea de longitud 3 hacia el este con una bolita de color azul en cada celda, empezando en la celda actual?
- ▶ ¿Es correcto el siguiente programa?

```
1  /* Proposito: Ver arriba
2  * Cabezal: queda en la celda inicial
3  * Precondicion: ???
4  */
5  program {
6      Poner(Azul); Mover(Este);
7      Poner(Azul); Mover(Este);
8      Poner(Azul); Mover(Este);
9      Mover(Oeste); Mover(Oeste); Mover(Oeste)
10 }
```

Regla de la precondition débil, ejemplo

- ▶ ¿Cuál debería ser la precondition de un programa que pone una línea de longitud 3 hacia el este con una bolita de color azul en cada celda, empezando en la celda actual?
- ▶ ¿Es correcto el siguiente programa?

```
1  /* Proposito: Ver arriba
2  * Cabezal: queda en la celda inicial
3  * Precondicion: ???
4  */
5  program {
6      Poner(Azul); Mover(Este);
7      Poner(Azul); Mover(Este);
8      Poner(Azul); Mover(Este);
9      Mover(Oeste); Mover(Oeste); Mover(Oeste)
10 }
```

- ▶ El programa correcto resuelve un problema con una precondition más débil.

¿Por qué documentar?

- ▶ Documentar: describir qué, cómo y por qué se hace algo

¿Por qué documentar?

- ▶ Documentar: describir qué, cómo y por qué se hace algo
- ▶ Programa: **texto** que describe cómo transformar estados
 - ▶ ¿Quién es el **destinatario** del texto?

¿Por qué documentar?

- ▶ Documentar: describir qué, cómo y por qué se hace algo
- ▶ Programa: **texto** que describe cómo transformar estados
 - ▶ ¿Quién es el **destinatario** del texto?
 - ▶ ¿Qué es más fácil de leer para el destinatario?

¿Por qué documentar?

- ▶ Documentar: describir qué, cómo y por qué se hace algo
- ▶ Programa: **texto** que describe cómo transformar estados
 - ▶ ¿Quién es el **destinatario** del texto?
 - ▶ ¿Qué es más fácil de leer para el destinatario?
- ▶ Conclusión: la documentación **es parte** del programa

Documentación

- ▶ Escribir un programa que ponga una bolita de color Rojo en el extremo Noroeste del tablero. Documentélo

Documentación

- ▶ Escribir un programa que ponga una bolita de color Rojo en el extremo Noroeste del tablero. Documentélo
- ▶ Escribir un programa que saque dos bolitas azules de la celda lindante al Norte de la celda inicial. Documentélo

Documentación

- ▶ Escribir un programa que ponga una bolita de color Rojo en el extremo Noroeste del tablero. Documentélo
- ▶ Escribir un programa que saque dos bolitas azules de la celda lindante al Norte de la celda inicial. Documentélo
- ▶ Conclusión: la documentación **es parte** del programa

Resumen

Lo que vimos en la presente clase

- ▶ Programar
 - ▶ Comunicar cómo resolver un problema en una computadora.
 - ▶ Receptores: computadora (ejecuta), humano (comprende y mantiene)
- ▶ Gobstones
 - ▶ Tablero, bolitas y cabezal
 - ▶ Programa: texto que describe transformaciones
 - ▶ Denotación de valores, comandos, `program`
- ▶ Documentación como parte del programa
 - ▶ Comunicación humano-humano
- ▶ Efecto y propósito de un programa
 - ▶ Equivalencia de programas y problemas
 - ▶ Efecto y propósito de un programa
 - ▶ Precondición de un problema
 - ▶ Qué vs. cómo