

**Fundamentos para programación y robótica**  
**Módulo 4+5 – Robótica**  
**Capítulo 1 – Proyectos y Robótica**

## Agenda

- El método de proyectos aplicados a la robótica
  - Parte1(hoy) y Parte2 (próxima clase)
- Estrategias para abordar problemas complejos
- Caso de estudio: Desafío Butiá 2011
- Ejercicio Práctico
- Robótica: Paradigmas
- Ejercicio Práctico (segunda iteración)

## El método proyectos como técnica didáctica y efectiva aplicado a la robótica

Lo que vamos a ver...

- Introducción
- En que consiste el método de proyectos
- Cómo se organiza
- Actividades y responsabilidades
- Evaluación
- Dificultades

## Proyectos: Introducción

- Estrategia didáctica
  - Difiere del enfoque tradicional (educación bancaria)
- Actualmente utilizado en muchos cursos de la Facultad de Ingeniería (ejemplo Butiá: robótica educativa)
- Utilizado en el mercado laboral
  - Desarrollo de software

## Proyectos: ¿En qué consiste?

- Propone que, a través del trabajo en proyectos, los estudiantes desarrollen y apliquen habilidades y conocimientos
- En base a un conjunto de experiencias atractivas los estudiantes se involucren en proyectos complejos
- Fomentar que los estudiantes tomen una mayor responsabilidad de su propio aprendizaje

Proyectos: ¿En qué consiste?

Características:

- Manejo de la información
- Manejar y usar en forma eficiente los recursos
  - Ej: tiempo, robots, tecnología
- Actividades sociales y personales
- Involucra a los estudiantes en la solución de problemas
- Les permite trabajar de manera autónoma

## Proyectos: ¿En qué consiste?

- Puede cambiar las relaciones entre los maestros y los estudiantes
- Permite reducir la competencia entre los alumnos y generar colaboración unos con otros
- Cambia el aprendizaje:
  - Lleva la simple memorización de hechos a la *exploración de ideas*

Proyectos: ¿En qué consiste?

Características que facilitan su implementación:

- Planearlo basado en un problema real
- Reducido o simplificado a un entorno académico
- Oportunidad para que los estudiantes realicen sus propias investigaciones
- Colaborar entre estudiantes, maestros y otras personas con el fin de compartir y distribuir el conocimiento
- Permite el uso de herramientas cognitivas y ambientes de aprendizaje que motiven al estudiante (ej: robot, laboratorio)

Proyectos: ¿En qué consiste?

Permite a los alumnos:

- Formar sus propias representaciones de tópicos y cuestiones complejas
- Determinar aspectos que encajan con sus habilidades e intereses
- **Trabajar en tópicos actuales relevantes y de interés local** (responsabilidad del docente)
- Las diferentes actividades involucradas permiten a los alumnos construir su conocimiento favoreciendo su retención y transferencia

## Proyectos: ¿En qué consiste?

Involucra tareas de investigación que generan en los estudiantes oportunidad de:

- Aprender ideas y habilidades complejas en escenarios realistas
- Aplicar habilidades en varios contextos
- Combinar habilidades completando
  - Tareas “expertas”
  - Deberes profesionales
  - Simulación de trabajo
- Resolver problemas

Proyectos: ¿En qué consiste?

Genera diversas aproximaciones al aprendizaje:

- Múltiples maneras en que los estudiantes demuestran su conocimiento
- Compatible con diferentes estilos de aprendizaje
- Permite alejarse de lo que hacen típicamente
  - Ej: Los que acostumbran ser seguidores pasan a ser líderes
- Desarrolla habilidades de colaboración y competencia en los estudiantes

## Proyectos: ¿En qué consiste?

### Prevenir y resolver conflictos

- Desarrollan habilidades sociales relacionadas al trabajo en grupo y la negociación
- Promueve asimilación de valores, conceptos y nuevas formas de pensamiento
- Invita a estudiantes a explicar o defender su posición ante los demás en sus proyectos

## Proyectos: ¿En qué consiste?

### Uso de la tecnología

- Expande la capacidad de los alumnos para presentar y manipular la información
- Se incrementan los intereses y las opciones profesionales de los alumnos
- Multiplica los medios en que los estudiantes pueden contribuir, como individuos, al proyecto
- En particular, el trabajo con robots abre un gran abanico de áreas en las cuales los estudiantes pueden participar

## Proyectos: Ejercicios

Trabajando en grupos de 3 personas:

- Plantear 4 problemas actuales, simplificados, de forma que puedan ser llevados adelante por estudiantes en un ambiente académico.
- Ejemplo:
  - *Título: Barrera de un peaje*
  - *Descripción: El problema consiste en realizar una barrera **autónoma** que detecte cuando un auto llega a la entrada, levante la barrera y una vez que el auto pasó, baje la barrera.*

## Estrategias para abordar problemas complejos

- Divide and Conquer
- Top  $\leftrightarrow$  Down
  - Top Down (“De Arriba hacia Abajo”)
  - Bottom Up

## Estrategias para abordar problemas complejos

- Divide and Conquer

## Estrategia Divide & Conquer (Divide y Vencerás)

- Uno de los paradigmas más importantes del diseño de algoritmos.
- Implica resolver un problema difícil descomponiéndolo en partes más simples, tantas veces como sea necesario.
- La resolución al problema difícil se realiza integrando cada una de las soluciones encontradas.

## Estrategia Divide & Conquer (Divide y Vencerás)

- Basado en la resolución recursiva de un problema complejo.
- Las “hojas” de la recursión buscan ser problemas cuya resolución sea “obvia”.
- Nota: Un algoritmo recursivo es aquel que expresa la solución en función de llamadas a sí mismo.

## Estrategia Divide & Conquer

### Algoritmo

- **DividirProblema (problema: Problema):**
  - if (esSimple(problema)):
    - resolver(problema)
  - else
    - sol1 = dividirProblema (getSubProblema(problema))
    - sol2 = dividirProblema (getrestoSubProblemas(problema))
    - return integrarSoluciones (sol1, sol2)

## Estrategia Divide & Conquer

Ejemplo: Llegar desde mi casa hasta el salón de clases en la Facultad de Ingeniería

### 1) Divido en subProblemas

Salir de mi casa

Transportarme a la facultad

Llegar al salón de clases

### 2) Analizo cada subProblema nuevamente

## Estrategia Divide & Conquer

Ejemplo: Hacer un programa interactivo que calcule el área de un triángulo

### 1) Divido en subProblemas

- Obtener los lados del triángulo

- Calcular el área

- Mostrar el resultado

### 2) Analizo cada subProblema nuevamente

## Estrategia Divide & Conquer

Ejemplo:

Obtener los datos del triángulo

- Le pido al usuario la base y la altura del triángulo

Calculo el área

- $(\text{Base} \times \text{Altura}) / 2$

Mostrar el resultado

- Imprimo el resultado en pantalla

## Estrategia Divide & Conquer

Ejemplo:

Obtener los lados del triángulo

- `alt = raw_input("ingrese altura: ")`
- `base = raw_input("ingrese base: ")`

Calculo el área

- `area = (alt*base)/2`

Mostrar el resultado

- `print "el area es ", area`

## Proyectos: Ejercicios

Trabajando en grupos de 2 personas:

- Descomponer los siguientes problemas utilizando la técnica Divide and Conquer
  - Realice robot seguidor de líneas
  - Realice una barrera para el control de autos en un centro comercial
  - Genere el comportamiento de un robot aspiradora

## Estrategias para abordar problemas complejos

- Top Down
- Bottom Up

## Estrategias Top Down y Bottom Up

- Diferentes enfoques utilizados principalmente en programación sobre cómo abordar la solución a un problema
- El enfoque Top Down aborda el problema desde un enfoque general y va refinando la solución en cada subnivel.
- El enfoque Bottom Up por el contrario, resuelve primero problemas concretos y luego realiza la integración hasta obtener la solución general.

## Estrategia Top Down

- El enfoque Top Down busca un completo entendimiento del problema a resolver antes de comenzar a “programar”
- Vamos desde enunciados abstractos, pasando por subniveles hasta obtener la solución concreta y detallada del problema

## Estrategia Top Down

Ejemplo: Calcular el área del triángulo

Calcular el área del triángulo



Multiplicar Base x Altura y dividir entre dos



$area = (base * altura) / 2$

## Enfoque Bottom Up

Se centra en especificar en gran detalle los elementos básicos de un sistema.

Estos elementos son más tarde combinados para obtener un sistema más complejo.

Cuando hay tecnologías de por medio el Top Down falla y es conveniente utilizar Bottom Up o una combinación de ambos.

## Enfoque Bottom Up

Ejemplo: Robot seguidor de líneas

- 1) Sensar diferentes colores en el suelo
- 2) Mover un motor

# Caso de estudio

## Desafío Butiá Básico 2011

- **Caso de estudio: Desafío Butiá Básico 2011**
  - Propuesta: El robot deberá ser capaz de seguir una línea negra dibujada en el suelo y además esquivar obstáculos que puede haber a lo largo del recorrido (video presentación).

## Caso de estudio: Desafío Butiá Básico 2011

- Aspectos a tener en cuenta del robot
  - Mecánico
  - Eléctrico
  - Programación
- Otros aspectos
  - Cantidad de personas
  - Cantidad de tiempo disponible para realizarlo

## Caso de estudio: Desafío Butiá Básico 2011

- Mecánica - Discusión
  - ¿Qué tipo de sensores vamos a utilizar?
  - ¿Cuántos de cada tipo?
  - ¿Dónde vamos a colocar los sensores?
  - ¿Qué fichas vamos a usar para afirmar los sensores?

## Caso de estudio: Desafío Butiá Básico 2011

- Eléctrica - Discusión
  - Tener claro como encender el robot
  - Pilas cargadas
  - Revisar que los cables estén conectados
  - Cables para los sensores
  - ¿Los motores están funcionando bien?

## Caso de estudio: Desafío Butiá Básico 2011

- Programación - Discusión
  - Seguidor de líneas
    - Avanzar sobre la línea
    - Buscar cuando la perdí
  - Esquivar obstáculos
    - Detectar obstáculos
    - Maniobra evasiva
  - Adelantándonos...comportamiento Reactivo

## Caso de estudio: Desafío Butiá Básico 2011

### Otros aspectos - Discusión

- ¿Cantidad de estudiantes?
- ¿Tiempo necesario para realizarlo?

## Caso de estudio: Desafío Butiá Básico 2011

- Video de la solución
- ¿Qué aspectos creen que se le pueden mejorar?

## Caso de estudio: Desafío Butiá Básico 2011

- Cantidad y tipo de sensores
- Ubicación de los mismos
- Comportamiento evasivo
- Comportamiento de seguir la línea

- Ejercicio Práctico:

- Trabajando en grupos de 4 personas y aplicando todos los métodos vistos hasta el momento resolver el siguiente problema:

*Se necesita realizar un robot aspiradora que se mueva libremente en un entorno en el cual puede haber obstáculos y “precipicios”. El robot debe ser capaz de evitar obstáculos y precipicios. Además deberá recorrer el entorno intentando cubrirlo en su totalidad.*

## Ejercicio Práctico:

Para resolver este ejercicio vamos a seguir la metodología vista:

- Aplicaremos la estrategia Divide&Conquer junto con el enfoque Top Down.
- ¿Qué aspectos debemos considerar? - Discusión

## Ejercicio Práctico:

Separaremos al robot en sus tres partes básicas

- Mecánica (recordar el módulo de mecánica)
  - Construcción
  - Forma
  - Distribución elementos clave
- Eléctrica
  - Conexiones
  - Baterías
- Programación
  - Alto nivel
  - Rutinas principales
  - Definir recorrido del robot

- Robótica

# Paradigmas

- Robótica: Paradigmas

**Paradigma:** Ejemplo o ejemplar. Modelo.

Los paradigmas nos sirven de modelos a la hora de desarrollar una solución robótica.

- Robótica: Paradigmas
- Existen tres paradigmas para organizar la inteligencia en un robot:
  - Jerárquico
  - Reactivo
  - Híbrido Deliberativo/Reactivo
- Existen tres funciones primitivas en robótica:
  - Sensar (SENSE)
  - Planificar (PLAN)
  - Actuar (ACT)

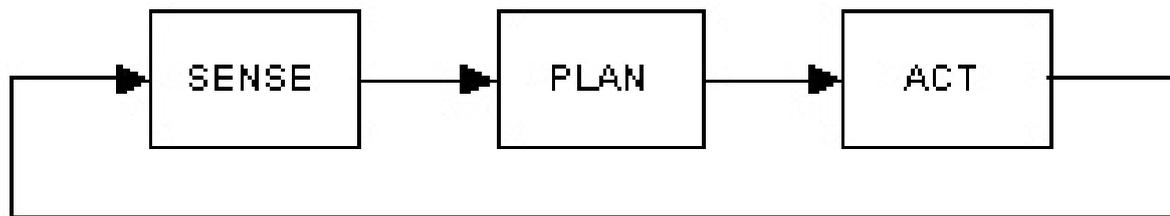
- Robótica: Paradigmas

## Primitivas robóticas

Primitiva robótica	Entrada	Salida
Sensar (SENSE)	Datos de los sensores	Información sensada
Planificar (PLAN)	Información (sensorial o cognitiva)	Directivas
Actuar (ACT)	Información sensada o directivas	Comandos a los actuadores

- Robótica: Paradigmas
- Los paradigmas se describen de dos maneras:
  - Por la relación entre las primitivas SENSAR, PLANIFICAR y ACTUAR.
  - Por la manera en que los datos sensoriales son procesados y distribuidos en el sistema.

- Robótica: Paradigmas
- Jerárquico (1967-1990)
  - Está basado en una visión introspectiva de cómo las personas piensan.
  - En cada paso se planifica qué hacer.



- Se arma un modelo global del mundo el cual es utilizado para planificar las acciones.

## Robótica: Paradigmas

- Primer robot IA.
- Desarrollado por el Stanford Research Institute (SRI) para DARPA 1967-9.
- Usa STRIP para determinar que acción tomar.
- Costo u\$s 100.000.
- Le fue bastante mal :S



# Fundamentos para programación y robótica

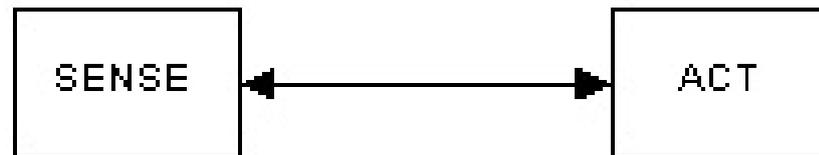
Primitiva robótica	Entrada	Salida
Sensar (SENSE)	Datos de los sensores	Información sensada
Planificar (PLAN)	Información (sensorial o cognitiva)	Directivas
Actuar (ACT)	directivas	Comandos a los actuadores

```
graph TD; S[Información sensada] --> P[Información (sensorial o cognitiva)]; P --> A[directivas];
```

## Robótica: Paradigmas

- Reactivo (1988-1992)

- Este paradigma aparece de los estudios en biología y psicología cognitiva.
- Elimina totalmente la planificación.



- Rápida ejecución al eliminar la etapa de planificación.
- Es el paradigma en el cual nos centraremos

## Robótica: Paradigmas

Primitiva robótica	Entrada	Salida
Sensar (SENSE)	Datos de los sensores	Información sensada
Actuar (ACT)	Información sensada 	Comandos a los actuadores



The diagram illustrates the flow of information in a robotic paradigm. It consists of a table with three columns: 'Primitiva robótica', 'Entrada', and 'Salida'. The first row shows 'Sensar (SENSE)' as the primitive, with 'Datos de los sensores' as the input and 'Información sensada' as the output. A horizontal arrow points from the input to the output. A second row is completely blacked out. The third row shows 'Actuar (ACT)' as the primitive, with 'Información sensada' as the input and 'Comandos a los actuadores' as the output. A horizontal arrow points from the input to the output. A diagonal arrow points from the 'Información sensada' output of the first row to the 'Información sensada' input of the third row. There is a black redaction box under the 'Información sensada' input of the third row.

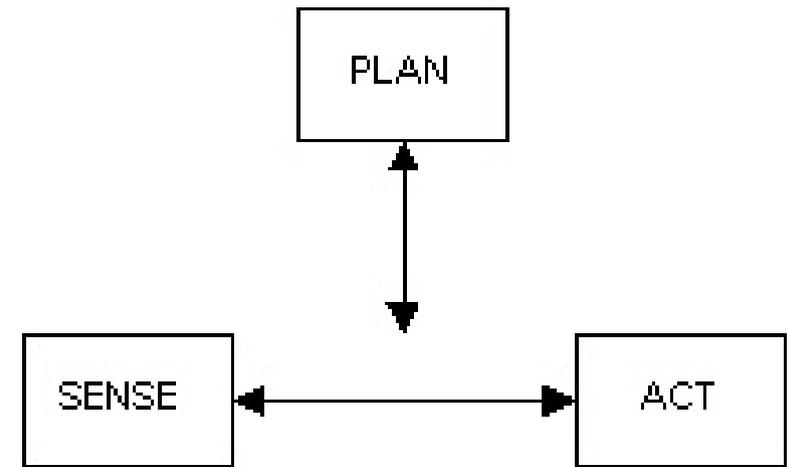
## Robótica: Paradigmas

- Ejercicio:
  - ¿En qué se parece el paradigma reactivo a lo visto en módulos anteriores?
  - ¿Cuál sería la implementación de un robot que debe frenar al detectar obstáculos? (programación)
  - ¿Cuál sería la implementación para un robot que tiene que mantenerse en el doyho de sumo? (programación)

## Robótica: Paradigmas

### ● Híbrido (1990's)

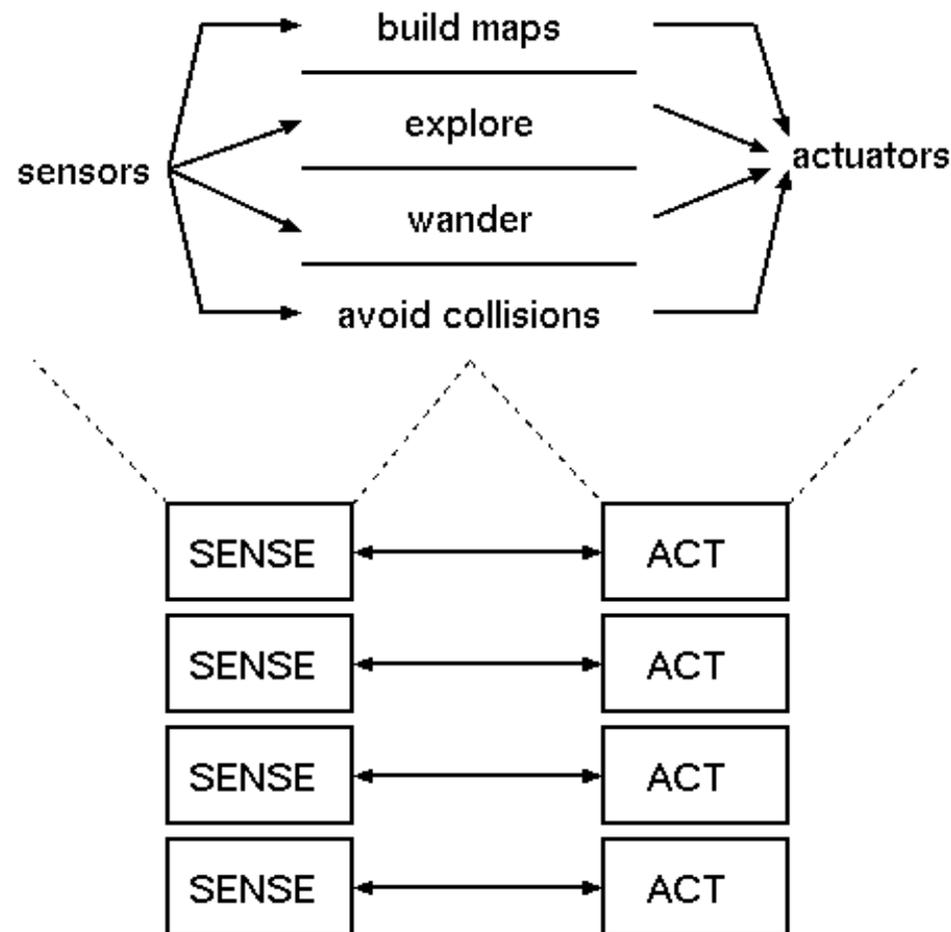
- Bajo este paradigma el robot primero planifica como descomponer la tarea en subtareas y luego cuales son los comportamientos adecuados para realizar las subtareas.
- Luego se ejecutan los comportamientos adecuados para cada subtarea.
- Es el que “utilizamos” los humanos
- Ejemplo: Aspiradora robot (navibot)



## Robótica: Paradigmas

### Descomposición Vertical

- Esta descomposición se asocia a los sistemas Reactivos.
- Comportamientos de bajo nivel asociados a instinto de supervivencia.
- Comportamientos de alto nivel para resolver tareas complejas.
- Es necesario definir un mecanismo para determinar la acción a tomar.



## Robótica: **Control basado en comportamientos**

- Inspiración desde la biología.
- Los componentes de estas arquitecturas son llamados comportamientos.
- Comportamiento, es un patrón de actividad que emerge de la interacción con el entorno.
- El sistema se construye de manera bottom-up.
- Los comportamientos pueden almacenar representaciones.

## Robótica: **Paradigmas en el tiempo**

- Reactivo
  - existe en el presente.
- Deliberativo
  - Puede razonar sobre el pasado.
  - Puede proyectarse en el futuro.

## Robótica: **Paradigmas**

### **Dilema pensar-reaccionar**

- Pensar lleva tiempo, mientras que reacción ejecuta rápido.
- Pensar permite planificar lo que evita tomar malas acciones, pero pensar mucho puede ser tan peligroso como tomar una mala acción.
- Para pensar se necesita una gran cantidad de información precisa. La cual debe mantenerse actualizada.
- El entorno es dinámico por lo que cuanto más se piense menos precisa la información y por ende la solución.

## Robótica: **Resumen**

- El control del robot es el proceso de tomar información del entorno, procesarlo de manera de tomar decisiones de actuación, y ejecutar esas acciones en el entorno.
- Las principales clases de controles son:
  - Control reactivo (no pensar, actuar)
  - Control deliberativo (pensar, luego actuar)
  - Control híbrido (pensar y actuar independientemente y en paralelo)
  - Control basado en comportamientos (pensar la manera de actuar)

## Robótica: **Resumen**

- Como ya dijimos, nos centraremos en el paradigma **Reactivo**.

- Ejercicio Práctico (segunda iteración):
  - Trabajando en grupos de 4 personas y aplicando todos los métodos vistos hasta el momento resolver el siguiente problema:

*Se necesita realizar un robot aspiradora que se mueva libremente en un entorno en el cual puede haber obstáculos y “precipicios”. El robot debe ser capaz de detenerse ante los obstáculos y ante los precipicios. Además deberá recorrer el entorno intentando cubrirlo en su totalidad.*

- Ejercicio Práctico (segunda iteración):
  - *Se propone:*
    - 1) *Cada grupo debe planificar el nuevo requerimiento (crear conocimiento)*
    - 2) *Luego, discusión entre los grupos (compartir el conocimiento y crear nuevo conocimiento)*
    - 3) *Cada grupo procede a la implementación luego del debate*