

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

MEMORIA TITULACIÓN

---

**Diseño y construcción de plataforma  
para estudios en enjambre de robots.**

---

*Supervisores:*

*Autor:*

Sebastián SÁEZ

Dra. María José ESCOBAR

Dr. Juan Cristobal ZAGAL

Dr. Pablo PRIETO

*Memoria para optar al grado de Bachiller en Ciencias de la Ingeniería y  
Licenciado en Ciencias de la Ingeniería Electrónica.*

*en el*

Síntesis de Máquinas Inteligentes  
Ing. civil Mecánica, Universidad de Chile

6 de junio de 2013

UNIVERSITY NAME (IN BLOCK CAPITALS)

## *Abstract*

Faculty Name  
Ing. civil Mecánica, Universidad de Chile

Bachiller en Ciencias de la Ingeniería y Licenciado en Ciencias de la Ingeniería  
Electrónica.

**Diseño y construcción de plataforma para estudios en enjambre de robots.**

por Sebastián SÁEZ

Desarrollo de una plataforma, de bajo costo, que permita estudiar el comportamiento de grupos de robots. Para que la plataforma funcione es necesario el diseño y construcción de los robots , implementar sistema de tracking 2D, armar la red de comunicación (XBee) y construir el software de control.

## *Acknowledgements*

The acknowledgements and the people to thank go here, don't forget to include your project advisor...

# Índice general

<b>Abstract</b>	<b>I</b>
<b>Acknowledgements</b>	<b>II</b>
<b>List of Figures</b>	<b>V</b>
<b>List of Tables</b>	<b>VI</b>
<b>Abbreviations</b>	<b>VII</b>
<b>Physical Constants</b>	<b>VIII</b>
<b>Symbols</b>	<b>IX</b>
<b>1. Robot</b>	<b>1</b>
1.1. Desarrollo tecnológico desde Chile . . . . .	1
1.2. ¿Qué es un robot? . . . . .	2
1.2.1. Automodelamiento . . . . .	3
<b>2. Swarm</b>	<b>6</b>
2.1. Comportamiento colectivo en la naturaleza . . . . .	6
2.2. Robots para construir un enjambre . . . . .	7
2.2.1. Kilobot . . . . .	7
2.2.2. Organismo Multibot . . . . .	8
2.2.3. E-puck . . . . .	9
2.2.4. 3pi Robot . . . . .	9
2.3. Necesidades de mercado . . . . .	11
2.3.1. Usos Académico . . . . .	12
2.3.2. Usos Militar . . . . .	12
2.3.3. Usos Doméstico . . . . .	12
<b>3. MODI</b>	<b>13</b>
3.1. Setup . . . . .	13
3.2. Construcción . . . . .	14
3.2.1. Fabricación Digital vs Análoga . . . . .	14
3.2.2. Software CAD . . . . .	18

3.2.3. Impresora 3D . . . . .	19
3.2.4. Diseño PCB . . . . .	22
3.3. Componentes electrónicos . . . . .	24
3.3.1. Arduino FIO . . . . .	25
3.3.2. Motores DC . . . . .	26
3.3.3. XBee . . . . .	27
 <b>A. Appendix Title Here</b>	 <b>28</b>
 <b>Bibliografía</b>	 <b>29</b>

# Índice de figuras

1.1.	Digesting Duck . . . . .	2
1.2.	Roomba y KUKA . . . . .	3
1.3.	Automodelado . . . . .	4
1.4.	Automodelado . . . . .	4
1.5.	AutomodeladoLIPSON . . . . .	5
2.1.	swarm . . . . .	6
2.2.	swarm . . . . .	8
2.3.	swarm . . . . .	8
2.4.	swarm . . . . .	9
2.5.	swarm . . . . .	10
2.6.	nicho . . . . .	11
3.1.	Setup . . . . .	13
3.2.	Setup . . . . .	14
3.3.	modi . . . . .	15
3.4.	modi2 . . . . .	15
3.5.	modirev1 . . . . .	16
3.6.	modirev1 . . . . .	16
3.7.	modirev1 . . . . .	17
3.8.	modirev1 . . . . .	18
3.9.	ModiSolidWorks . . . . .	19
3.10.	Render . . . . .	19
3.11.	3Dprint . . . . .	20
3.12.	3DTouch . . . . .	21
3.13.	uprint . . . . .	21
3.14.	Replicator1 . . . . .	22
3.15.	cables . . . . .	22
3.16.	MDX20 . . . . .	23
3.17.	pcbeagle . . . . .	23
3.18.	pcb . . . . .	24
3.19.	fio . . . . .	25
3.20.	MotorDC . . . . .	26
3.21.	MotorDC . . . . .	27

# Índice de tablas

2.1. Comparación de robots Kilobot, e-puck y 3pi.	11
3.1. Comparación impresoras 3D	20
3.2. Especificaciones Fio	26
3.3. Motor	26
3.4. Xbee	27

# Abbreviations

**LAH** List Abbreviations Here

# Physical Constants

Speed of Light     $c$    =    $2,997\ 924\ 58 \times 10^8$  ms<sup>-1</sup> (exact)

# Symbols

$a$	distance	m
$P$	power	W (Js <sup>-1</sup> )
$\omega$	angular frequency	rads <sup>-1</sup>

*For/Dedicated to/To my...*

# **Capítulo 1**

## **Robot**

### **1.1. Desarrollo tecnológico desde Chile**

Hacer Robots en Valparaíso no es una tarea sencilla, la falta de lugares especializados para comprar dificulta contar con los componentes necesarios para construir una máquina.

Al comienzo de este proyecto se trabajó en Santiago, en conjunto con la Universidad de Chile (UChile) bajo la tutela del Doctor Juan Cristóbal Zagal en el Laboratorio de Síntesis de Máquinas Inteligentes. Por parte de la Universidad Técnica Federico Santa María (USM), se trabajó con la Doctora María José Escobar, del Departamento de Electrónica y luego se incorporó el Doctor Pablo Prieto del Departamento de Diseño de Productos. El financiamiento para realizar este prototipo ha sido por parte de las dos Instituciones, Universidad Técnica Federico Santa María y Universidad de Chile.

Durante el proceso de desarrollo se tuvieron que hacer diversas compras de materiales, pero los lugares más recurrentes al momento de hacerlas fueron Olimex y Casa Royal. La primera es una empresa dedicada a traer productos para hacer prototipos y construir máquinas, la segunda cuenta con varios insumos básicos para trabajar en desarrollo de circuitos electrónicos. Ambas empresas se encuentran en Santiago, por lo que trabajar en esta ciudad es de gran ayuda para reducir los tiempos en desarrollo.

Luego de armar un primer robot básico funcional, con materiales disponibles en Santiago de Chile, se hizo una búsqueda de componentes en tiendas especializadas que permiten comprar por internet. Existen varias (ver anexo 1) pero las más importantes para este proyecto fueron: Sparkfun y Seeedstudio.

## 1.2. ¿Qué es un robot?

Un robot puede ser un software solamente o tener además una extensión física que le permita interactuar con la realidad y realizar tareas de forma autónoma. Etimológicamente, el término robot se le atribuye al dramaturgo checo Karel Čapek, que en su obra R.U.R en 1921 (Rossum's Universal Robots) utilizó la palabra “robotnik” para referirse a ayudantes artificiales. Luego fue el escritor Isaac Asimov (1920-1992) quien, gracias a su obra, difundió la palabra robótica haciendo referencia a la ciencia encargada de estudiar a los robots. La robótica contempla el estudio de al menos 6 áreas: La mecánica, la electrónica, el control automático, la física y la matemática.

En la historia hay varios intentos por construir estos ayudantes artificiales. A principios del siglo XVIII, Jacques de Vaucanson creó un autómata capaz de tocar la flauta, así como un pato mecánico que continuamente seguía su ciclo biológico.

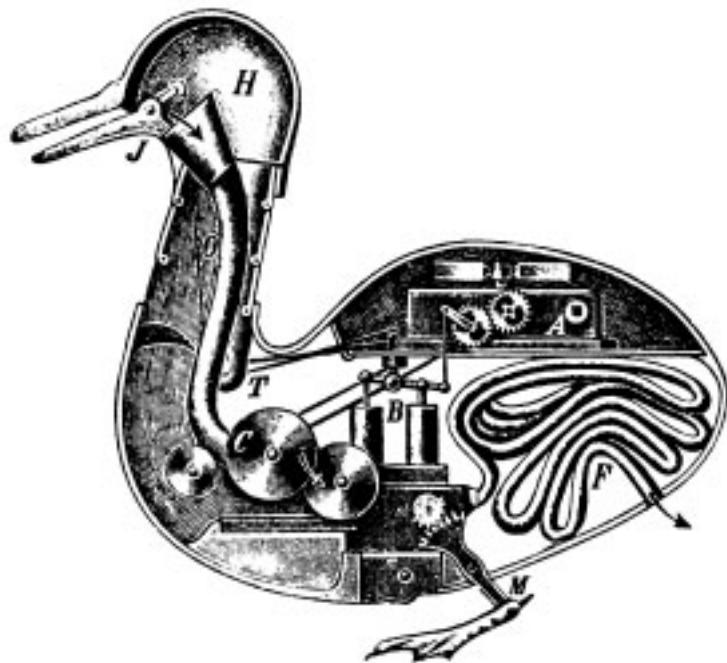


FIGURA 1.1: Digesting Duck, creado por Jacques de Vaucanson en 1739. Imagen extraída de Wikipedia.

En la actualidad las empresas KUKA, Honda y Sony, entre otras, construyen robots especialmente diseñados para la industria. Los robots que se utilizan en la industria, y los pocos que han llegado al hogar, son controlados por un algoritmo. Este es parte de un software que escribe una persona, donde se detalla la tarea que el robot debe realizar; tiene un modelo de los motores, partes y piezas para que así la máquina tenga información de como es, y pueda ejecutar la tarea para la cual se le programó. Si se interfiere con el entorno del robot, por ejemplo moviendo 1 [cm], fuera del rango de

los sensores, el perno que debe apretar algún robot industrial que ensambla autos, este no podrá “encontrarlo”. Los robots comerciales que existen hoy en día no son capaces de adaptarse a cambios en el entorno y menos ser capaces de generar una imagen de sí mismos que les permita entender qué sucede y recuperarse de fallas.



---

FIGURA 1.2: **Izquierda:** Robot Roomba, primer Robot doméstico vendido en chile, Imagen extraída de <http://www.irobotchile.cl>. **Derecha:** Robot industrial KUKA KR 1000 TITAM, Imagen extraída de <http://www.turbosquid.com>.

Un algoritmo debe tener un modelo detallado de los motores y sensores que posee el robot, y es tarea del programador hacer la abstracción necesaria para poder darle sentido al movimiento del conjunto de motores. Si hablamos de un robot de 4 extremidades, el programador debe ser capaz de indicar la secuencia de activación de cada motor para así primero hacer que el robot mueva una extremidad y luego con la suma de las 4 lograr desplazarse.

### 1.2.1. Automodelamiento

Juan Cristobal Zagal y Hod Lipson [1] exploraron el comportamiento de un robot capaz de entrar en un proceso de self-reflection. Ellos estudiaron a un robot al cual se le programaron dos controladores, uno primitivo (primitive controller) y uno reflexivo (reflective controller) que puede observar al primero.

El controlador reflexivo es capaz de determinar el control primitivo sin tener acceso directo a sus estados internos si no que haciendo uso de ingeniería inversa al leer los input/output de éste. Acá se implementa como comportamiento que el robot (e-puck) debe alejarse de luces rojas y acercarse a azules. Se logra con una exploración mínima posible de hardware.

El diagrama del algoritmo que controla todo el proceso que se implementa como comportamiento para recuperarse de fallas puede verse en el anexo.

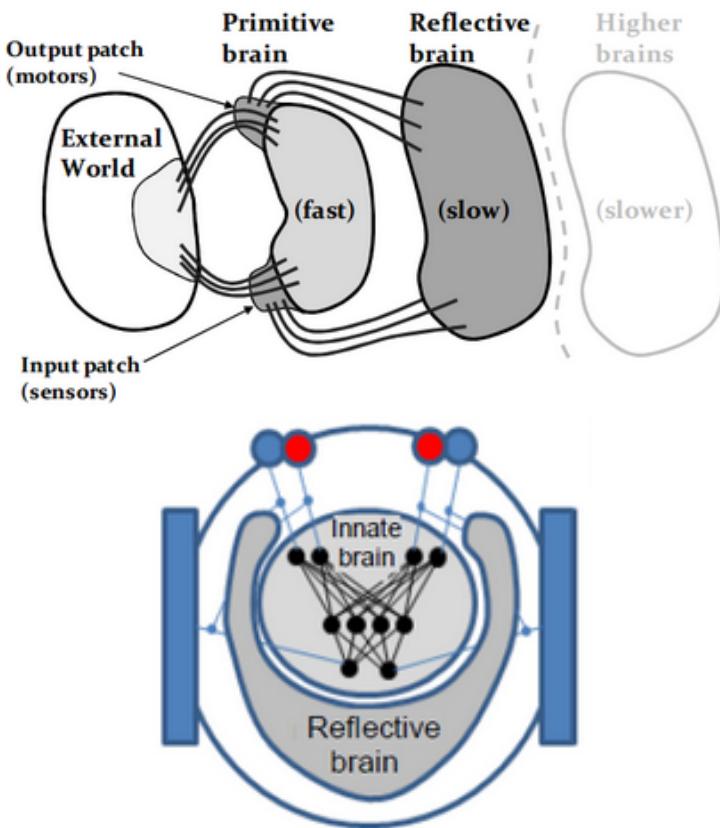


FIGURA 1.3: **Arriba:** Arquitectura de cerebros anidados usado en el trabajo de J.Zagal y H.Lipson, basado en modelos de M. Minsky. **Abajo:** Modelo de cerebros implementado en robot e-puck. Ambas imágenes pertenecen a Juan Cristóbal Zagal.

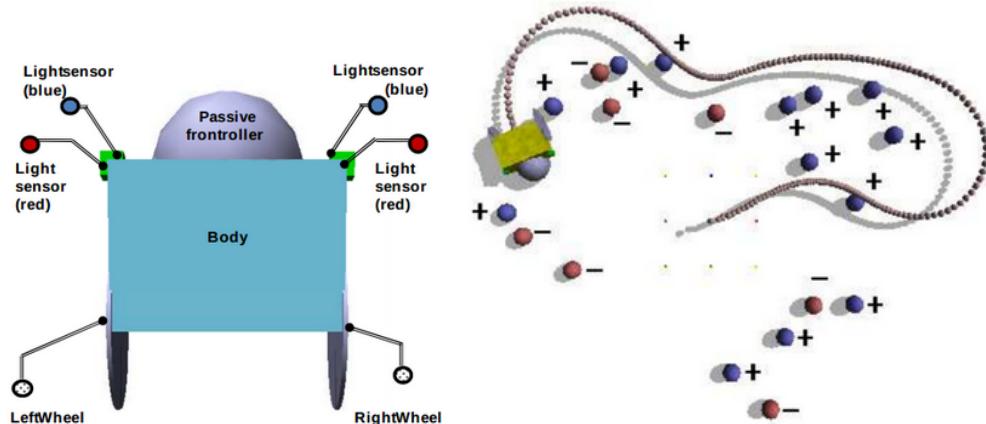


FIGURA 1.4: **Izquierdo:** Arquitectura de cerebros anidados usado en el trabajo de J.Zagal y H.Lipson, basado en modelos de M. Minsky. **Derecha:** Modelo de cerebros implementado en robot e-puck. Ambas imágenes pertenecen a Juan Cristóbal Zagal.

J. Bongard., V. Zykov y H. Lipson., logran que un robot se recupere de una falla inesperada de forma autónoma. Dicen que el robot puede recuperarse de manera autónoma haciendo uso de su (propio al robot) Self-modeling. Concretamente implementan sus algoritmos en un robot de 4 patas que utiliza una relación entre sus sensores y motores para indirectamente inferir su propia estructura.

Si se remueve una extremidad, el robot es capaz de generar una nueva forma de caminata que le permita cumplir con su misión que es avanzar. En la figura 1.5, se describe un esquema del algoritmo. El robot realiza una acción física (A), al azar, luego se ejecuta la mejor acción que se encuentre en (C). A continuación genera varios auto-modelos que coincidan con las lecturas de los sensores obtenidas en (B). Aún no sabe cual es el modelo, por lo que en (C) genera varias acciones posibles que acotan la búsqueda de modelos. Despues de varios ciclos de (A) a (C), el modelo obtenido se utiliza en (E) para generar la secuencia de locomoción. La mejor secuencia de locomoción es probada físicamente en el robot. Se refina el modelo volviendo al paso (B) y en (D) pueden crearse nuevos comportamientos.

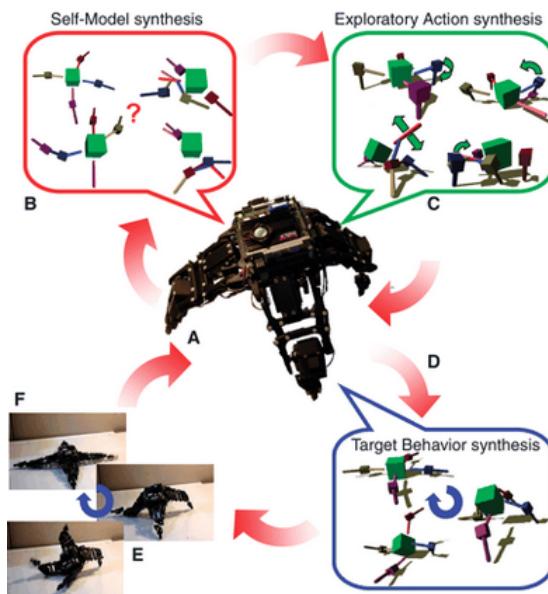


FIGURA 1.5: Esquema del algoritmo que puede usar un robot para desplazarse haciendo uso de Automodelos. Imagen obtenida desde <http://creativemachines.cornell.edu>

# Capítulo 2

## Swarm

### 2.1. Comportamiento colectivo en la naturaleza



---

FIGURA 2.1: Bandada de auklets, teniendo comportamiento de enjambre. Imagen extraída de Wikipedia.

Existen casos de enjambres, los más típicos son las hormigas y abejas, pero los hay en peces, aves e incluso los mamíferos. Son sistemas donde nadie está a cargo y aún así ejecutan una tarea grupal. Las hormigas son un gran ejemplo. Al momento de construir su nido, no tienen un arquitecto o ingeniero estructural que esté dando órdenes, simplemente cada una sabe que tiene que hacer. No hay un director orquestando la construcción desde lo alto, en vez de esto lo que ocurre es un comportamiento emergente. También conocido como inteligencia de enjambre.

Otro tipo de comportamiento colectivo son las migraciones, desplazamientos periódicos que efectúan aves, peces, langostas y mamíferos de un hábitat a otro. Cada individuo activo en la migración sigue al grupo, los más pequeños como el plancton o anfibios aprovechan las corrientes de aire o agua, y las aves, más grandes, aprovechan los vientos y corrientes ascendentes. Hay diversas finalidades detrás de la migración, algunas especies lo hacen para escaparse de los crudos inviernos o secos veranos; mientras que otras, como las tortugas marinas, por una necesidad reproductiva emprenden un largo viaje de más de 10.000 millas, a lo largo de todo el Atlántico Norte.

Lograr que un enjambre de robots tenga un comportamiento emergente como el de las colonias de abejas es la piedra filosofal de los investigadores de esta área. Uno de los más destacados investigadores del área, James McLurkin, experto en robótica del Massachusetts Institute of Technology (MIT) dice que para lograrlo es necesario un software que ejecute tareas individuales y que de alguna forma se cumpla con una tarea grupal. He aquí una importante razón para desarrollar estudios sobre enjambres de robots, ya que aún no está claro cómo se coordina la naturaleza para llevar a cabo tales tareas.

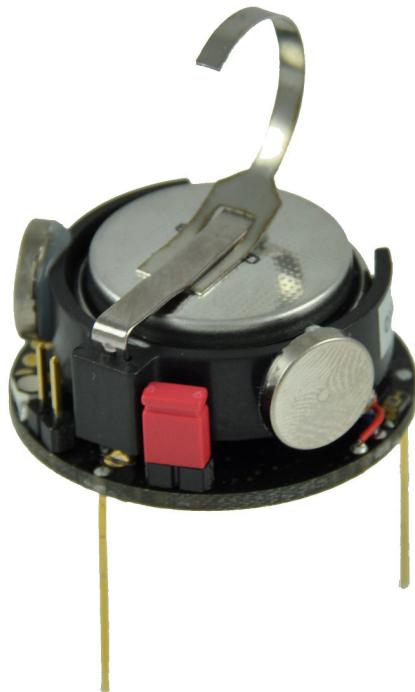
## 2.2. Robots para construir un enjambre

Imaginemos una situación hipotética donde un edificio es destruido, la búsqueda de sobrevivientes no es una tarea fácil, implica que rescatistas ingresen al lugar corriendo grave peligro, usualmente buscando a las víctimas en condiciones de poca visibilidad. Esto mismo podría ser ejecutado por un enjambre robótico que esté programado para buscar gente y que de manera colectiva recorra un área mucho mayor que 2 o 3 personas. Incluso un robot del mismo enjambre puede fallar, pero al ser un sistema distribuido el enjambre continúa funcionando, es un sistema muy robusto.

A continuación algunos robots que pueden ser utilizados para hacer un Robot Swarm junto con las información técnica disponible.

### 2.2.1. Kilobot

El proyecto Kilobot, es un sistema de bajo costo escalable para demostrar comportamientos colectivos. Actualmente existen varios grupos que están investigando algoritmos para enjambres de robots, por esto que diseñaron Kilobot que es un robot de bajo costo, accesible, que permite hacer pruebas en cientos o miles de robots.



---

FIGURA 2.2: Kilobot. Imagen extraida de <http://www.k-team.com>.

### 2.2.2. Organismo Multibot

S. Kornienko, O. Kornienko, A. Nagarathinam y P. Levi., exploran el trabajo colaborativo en robots para un mejor rendimiento y mayor fiabilidad a nivel macroscopico. En este articulo demuestran sus últimos trabajos en sistemas colectivos y lo más sorprendente es que logran la agregación y desagregación autónoma para así obtener un organismo multibot. [2]



---

FIGURA 2.3: Organismo Multibot. Imagen extraida de <http://ieeesbcet.org/>

### 2.2.3. E-puck

Uno de los robots más utilizados por los científicos en el mundo para estudios y publicaciones es el E-puck. Este robot es compacto, tiene forma de cilindro con un diámetro de 7 [cm] y para moverse hace uso de sus dos ruedas, dejándolo en la categoría de robot con desplazamiento diferencial. Originalmente fue diseñado para educar en el área de la micro ingeniería por Michael Bonani y Francesco Mondala en el laboratorio ASL del Profesor Roland Siegwart en Escuela Politécnica Federal de Lausana (EPFL) en Suiza. El e-puck es open hardware, software es de código abierto, lo construyen y venden varias empresas. Para comunicarse con una computador incorporan un módulo Bluetooth conectado a uno de sus dos puertos serie. Existen varios tipos de accesorios, entre los que destacan un Zigbee para comunicaciones, un módulo con varias cámaras y LEDs RGB como sistema de comunicación visual. Su precio a la fecha en Getronic es 912 USD. Para comprarlo hay que encargarlo desde Suiza.



---

FIGURA 2.4: E-puck. Imagen extraída de Wikipedia

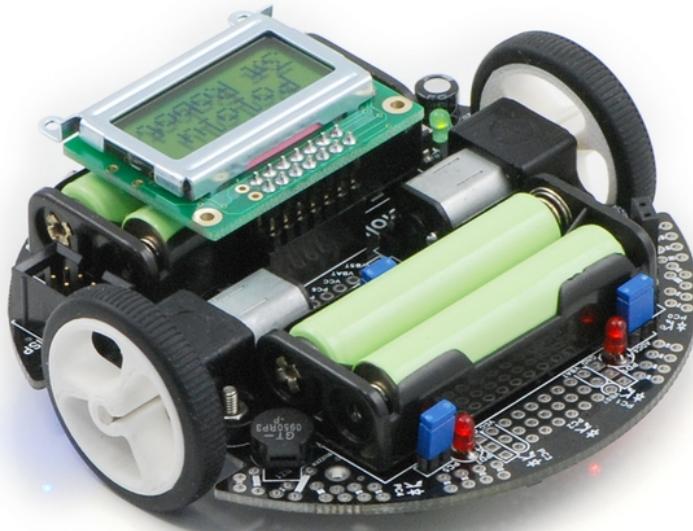
### 2.2.4. 3pi Robot

Pololu, la misma marca que tiene desarrollo de varios tipos de motores para robótica y PCB para controlarlos, diseño el 3pi Robot. Es un robot bastante más económico que el

e-puck, cuyo valor es 99.95USD ref. Sparkfun. También tiene dos ruedas para desplazarse de forma diferencial, 5 sensores de reflectancia, un LCD de 8x2 caracteres, un buzzer y tres botones para que el usuario pueda programarlos. Todos estos dispositivos están conectados a un microcontrolador ATmega328. Su velocidad es de 90 cm/s.

El 3pi fue diseñado especialmente como un robot seguidor de líneas y solucionador de laberintos. Existen varios videos que muestran la asombrosa velocidad de estos robots para solucionar un laberinto. Se programa en C, pero como posee un microcontrolador ATmega es posible hacer uso del bootloader Arduino y programarlo con ese IDE. Usa 4 baterías AAA y trae 4 LEDs.

Para el modelo que se planteó en este trabajo hace falta un módulo de comunicación inalámbrica, y el 3pi no tiene. Para usar esta alternativa es necesario sumarle a su precio 26USD, que es el precio del módulo XBee 1[w] serie 2 que se vende en Sparkfun.



---

FIGURA 2.5: 3pi Robot, simple y económico pero no tiene comunicación inalámbrica.  
Imagen extraída de <http://www.skpang.co.uk>

En la tabla ?? comparativa resume las principales características de los robots: Kilobot<sup>1</sup>, E-puck<sup>2</sup> y 3pi<sup>3</sup>.

---

<sup>1</sup><http://www.k-team.com/mobile-robotics-products/kilobot/specifications>

<sup>2</sup><http://www.getronic.com/doc/index.php/E-Puck>

<sup>3</sup><http://www.pololu.com/catalog/product/975/specs>

	Kilobot	E-puck	3pi
Procesador	ATmega 328 @ 8MHz	dsPIC 30 CPU @ 30 MHz	ATMega328
Memoria	32 KB Flash	114 KB Flash	32 KB Flash
Autonomía	3 meses (modo sleep)	3 horas	45 minutos
Cargador	opcional	opcional	opcional
Comunicacion	Infrarrojo(IR)	Bluetooth	opcional
Sensores	Potencia recibida IR	Proximidad, camara, otros	5 Reflectancia
Movimiento	2 motores vibración	2 motores diferencial	2 motores diferencial
Output	1 (RGB) LED	8 (RGB) LED + parlante	LCD 8x2 + buzzer
Diametro	33 [mm]	70 [mm]	95[mm]
Alto	34[mm]	50 [mm]	32[mm]
Software	Básico	Complejo	Básico
Programación	WinAVR	MPLAB	Arduino
Precio USD	1.161 (10 pack)	900	100

TABLA 2.1: Comparación de robots Kilobot, e-puck y 3pi.

### 2.3. Necesidades de mercado

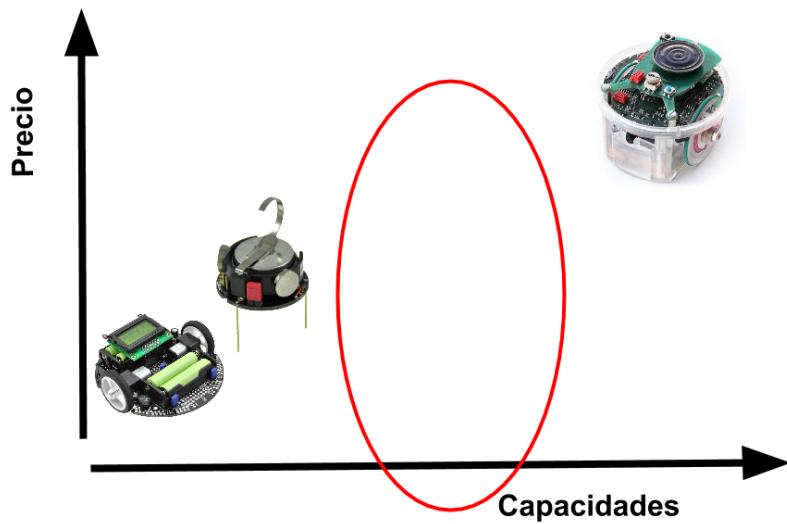


FIGURA 2.6: Nicho de mercado

De los robots estudiados destaca en sus prestaciones el e-puck, pero este tiene dos grandes problemas para ser usado por gente que no es especialista en robots. Uno es que tiene demasiado hardware, lo que tiende a confundir y aumentar costos. Dos, que para su comunicación inalámbrica hace uso de Bluetooth, protocolo que no soporta las redes Mesh para hacer de manera simple el control de muchos dispositivos en una red. Hace falta un robot capaz de controlarse de forma inalámbrica, simple de construir y fácil de usar. Además debe ser económico para poder construir varios. Se pretende hacer uso de tecnologías como impresoras 3D y de desarrollo como el Open Hardware.

### **2.3.1. Usos Académico**

Un enjambre de robots puede presentar muchas ventajas dentro del aula. Si se tiene un sistema de fácil uso para los alumnos, el profesor puede asignar una tarea a un grupo de estudiantes donde cada uno tiene la responsabilidad de controlar o programar un robot para que el conjunto logre una meta determinada como ordenar unos bloques o hacerse cargo de regar un pequeño huerto. Abusando un poco del concepto de la colectividad, incluso pueden generarse tareas donde cada colegio se especializa en un tipo de tareas para luego juntar los distintos robots y probar cómo interactúan.

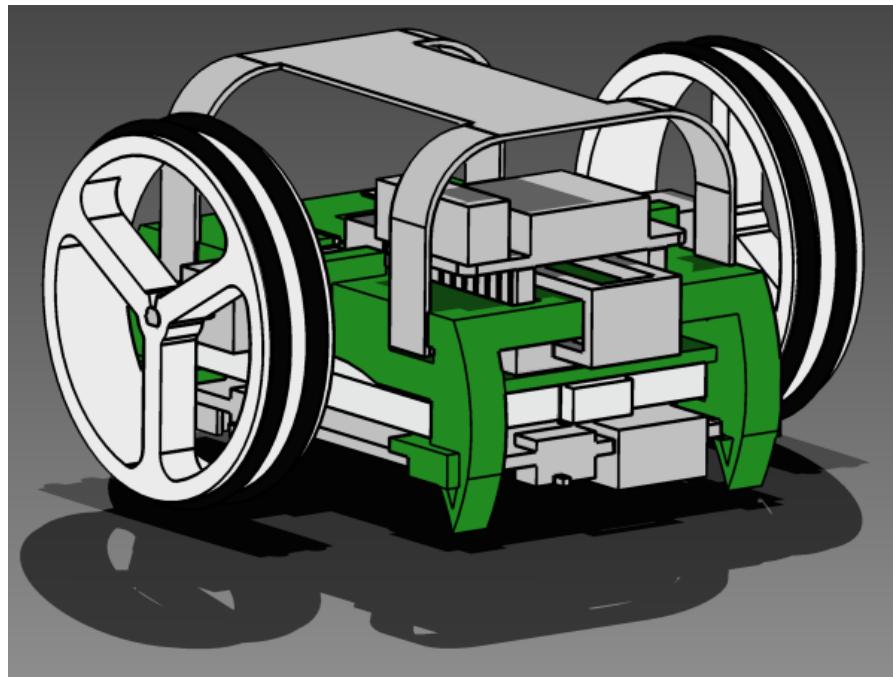
Tener un setup con robots que demuestren un comportamiento colectivo puede ser muy ventajoso para ayudar a niños con trastornos como el Asperger a practicar sus habilidades para reconocer estos mismo comportamientos.

### **2.3.2. Usos Militar**

### **2.3.3. Usos Doméstico**

# Capítulo 3

## MODI



---

FIGURA 3.1: robot MODI (sigla para Modular Intelligence)

### 3.1. Setup

Se desea realizar una investigación sobre el comportamiento colectivo de un grupo de robots, en el mundo real sin hacer uso de simuladores. Por esto es necesario contar con un lugar físico donde poder activar los robots. Además para simplificar cada uno de los robots, estos no tienen sensores internos por lo que hay una cámara montada sobre el plano de movimiento de estos, para hacer Seguimiento Visual.

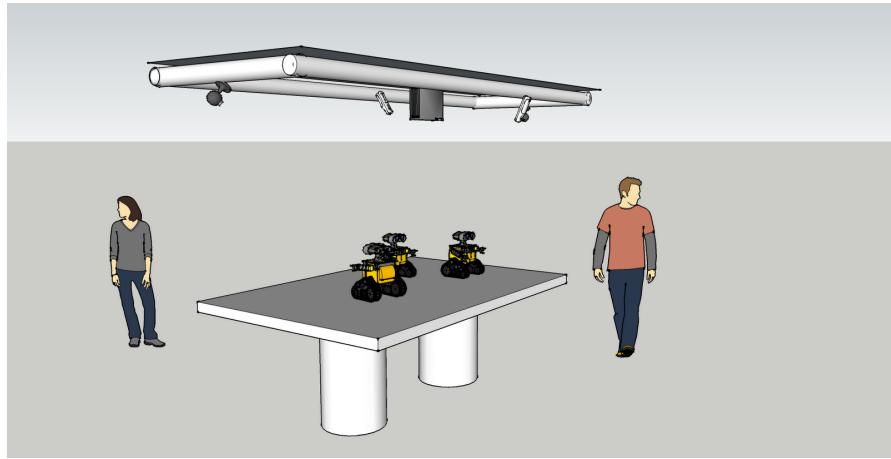


FIGURA 3.2: Setup a montarse para hacer estudios de grupos de robots.

La función principal de MODI es ser una plataforma móvil de fácil acceso. Existe un repositorio en GitHub donde se tienen los códigos actualizados para controlar y construir robots MODI. Para descargar códigos y piezas de construcción se debe ir a <https://github.com/FabLabUCh/modi>

### Las funciones principales que se desarrollaron

- Carga Autónoma con celda solar.
- Seguimiento de grupo.
- Control individual del color de cada MODI.
- Movimiento simple de cada robot de forma independiente.

## 3.2. Construcción

### 3.2.1. Fabricación Digital vs Análoga

Cuando se quiere pasar una idea al mundo real es necesario un proceso de fabricación. Dependiendo de la cantidad de herramientas que se tenga es más o menos fácil la tarea. Desde el comienzo hasta hace un par de años, quienes se dedican a construir robots, debían construir de manera “artesanal” donde es imposible que las piezas queden todas iguales y el tiempo empleado era bastante. Hoy en día existe una gran alternativa que surge como un nuevo paradigma, la Fabricación Digital. Las impresoras 3D, que no son más que un extrusor montado en un sistema con ejes que le dan 3 grados de libertad, permiten desde un modelo hecho en un computador, obtener un objeto real en plástico.

También existe otro tipo de máquinas que permite hacer diseños en 2D, estas son las cortadoras LASER. La primera versión de MODI fue construido usando planchas de madera MDF y acrílico, cortados en LASER.

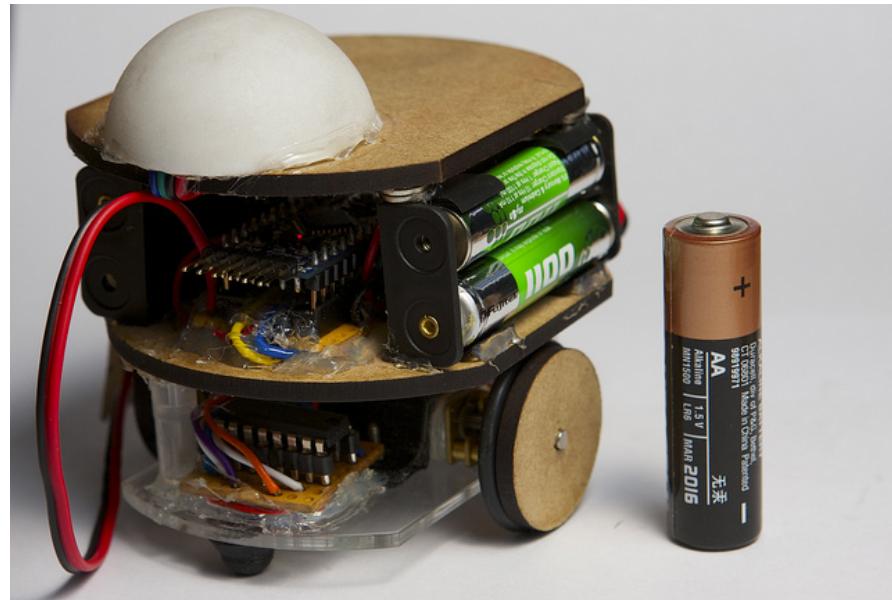


FIGURA 3.3: Primera versión robot MODI, tiene un Arduino mini pro, Xbee, usa pilas AAA y parte del chasis es de Madera MDF de 3[mm] cortado con LASER

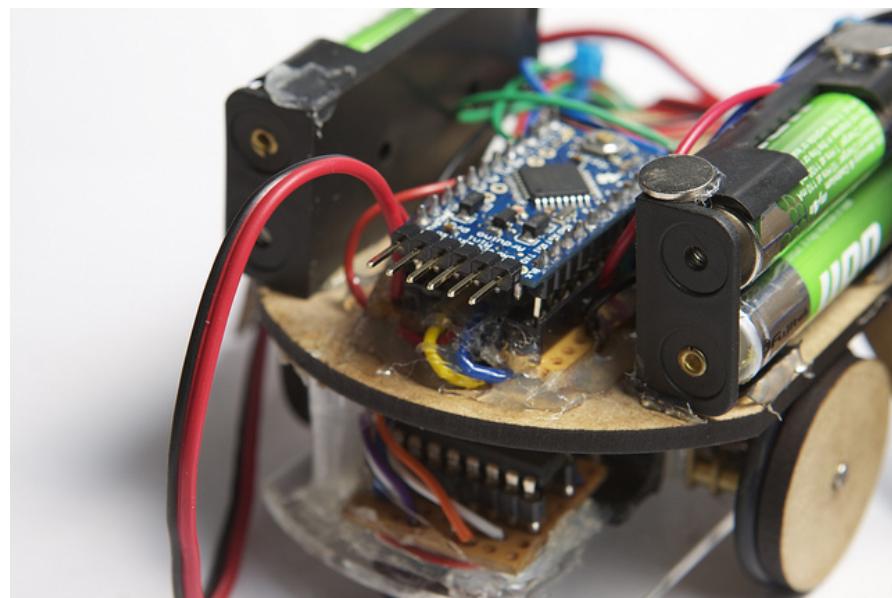


FIGURA 3.4: MODI, primera versión basada en gran parte en Fabricación Análoga.

Utilizando una impresora 3D Makerbot Replicator 1 se hizo esta primera versión con piezas plásticas.

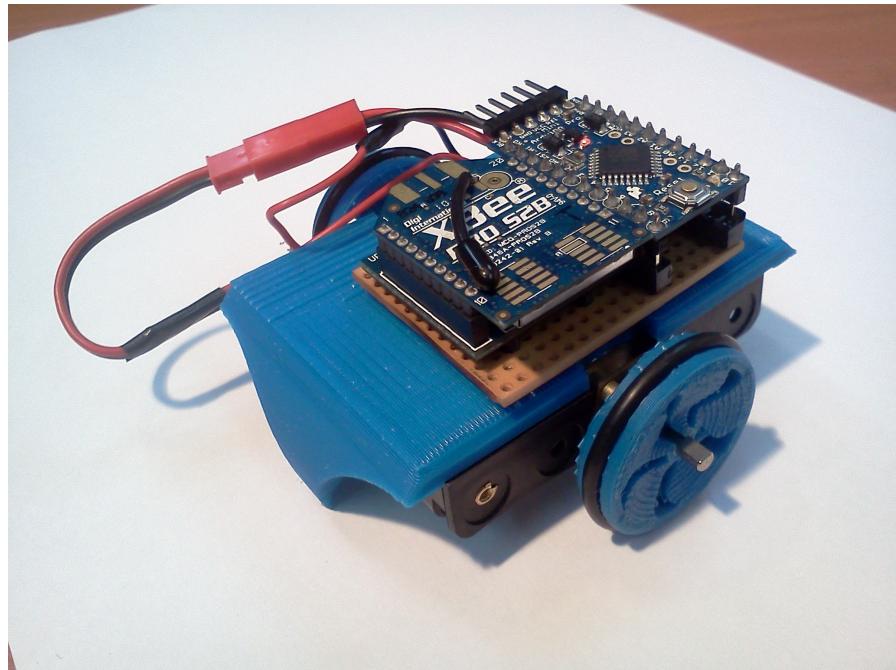


FIGURA 3.5: Primera versión de MODI usando técnica de *Fabricación Digital* con chasis de plástico construido con una MakerBot Replicator 1

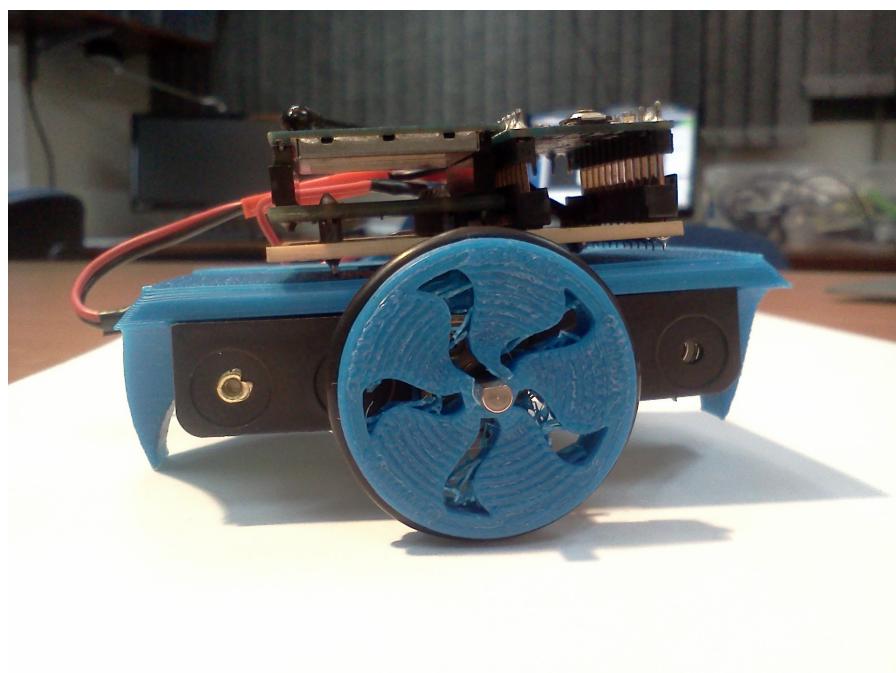


FIGURA 3.6: Vista lateral.

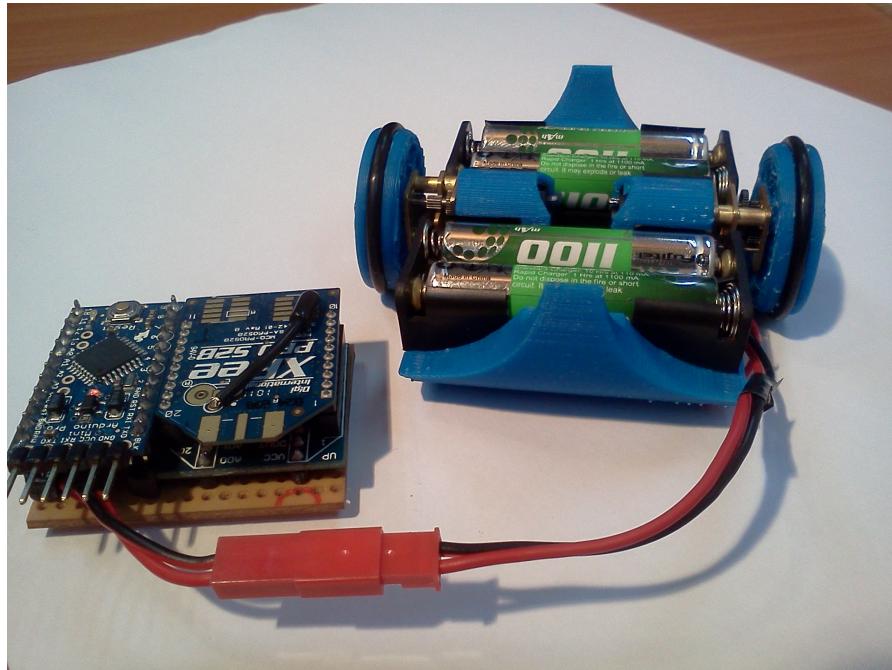


FIGURA 3.7: Vista inferior.

Esta versión se dio de baja ya que presentaba el problema de involucrar demasiadas partes que debían ser hechas por una persona. El microcontrolador, junto con la radio inalámbrica se colocaron en una placa electrónica para prototipado, más adelante se puede ver que fueron reemplazados por una PCB llamada Arduino FIO, que es una plataforma de desarrollo Arduino junto con un socket XBee. Otro factor clave para descartar esta versión es que utiliza 4 pilas AAA que necesitan ser removidas para poder ser recargadas, lo que impide que en futuras versiones exista la posibilidad de una carga autónoma por parte de los Robots. La versión actual de MODI permite su carga por medio de un puerto Mini USB, panel solar y de forma inalámbrica.

Aunque han bajado los precios de las máquinas para prototipado rápido, aún no están al alcance de todas las personas. Es por esto que existen los Fab Labs (acrónimo del inglés Fabrication Laboratory), que según Wikipedia es, *un espacio de producción de objetos físicos a escala personal o local que agrupa máquinas controladas por ordenadores. Su particularidad reside en su tamaño y en su fuerte vinculación con la sociedad*. Los Fab Labs están por todo el mundo, Figura 3.8. MODI fue concebido como un proyecto del Fab Lab de la Universidad de Chile y por esto es posible reproducirlo en cualquier Fab Lab. En Chile, además del Fab Lab de la Universidad de Chile están: Design LAB UAI, Fab Lab Santiago y Stgo MakerSpace.



FIGURA 3.8: Mapa actual de lugares en el mundo que cuentan con un FabLab. En Chile a la fecha existen 3. Imagen obtenida desde <http://fablabamersfoort.nl/fablabs/>

### 3.2.2. Software CAD

CAD viene de sus siglas del Inglés, Computer-aided design, y se refiere a un diseño asistido por herramientas computacionales. Profesionales como ingenieros, arquitectos y del área del diseño por lo general son los que hacen más uso de estas herramientas.

Según Wikipedia, "... se pueden dividir básicamente en programas de dibujo en dos dimensiones (2D) y modeladores en tres dimensiones (3D). Las herramientas de dibujo en 2D se basan en entidades geométricas vectoriales como puntos, líneas, arcos y polígonos, con las que se puede operar a través de una interfaz gráfica...."

Durante el transcurso del proyecto se trabajó con varios softwares CAD. El primero fué SketchUp 8 de Google, que permite fácilmente hacer bocetos de lugares y cuenta con una importante biblioteca de modelos para incluir en el diseño. Rápidamente se pudo hacer un sketch utilizando modelos descargados de Internet , Figura 3.2.

El diseño del chasis junto con las ruedas y demás partes plásticas, se hizo en un comienzo con SolidWorks 2012 y luego por ser más simple de usar, Inventor 2013 de Autodesk. Ambos softwares permiten generar modelos en 3D para luego exportar el diseño al formato STL que es estándar para prototipar en plástico.

Luego de varias iteraciones se logró este modelo, Figura 3.10 que tiene cinco piezas.

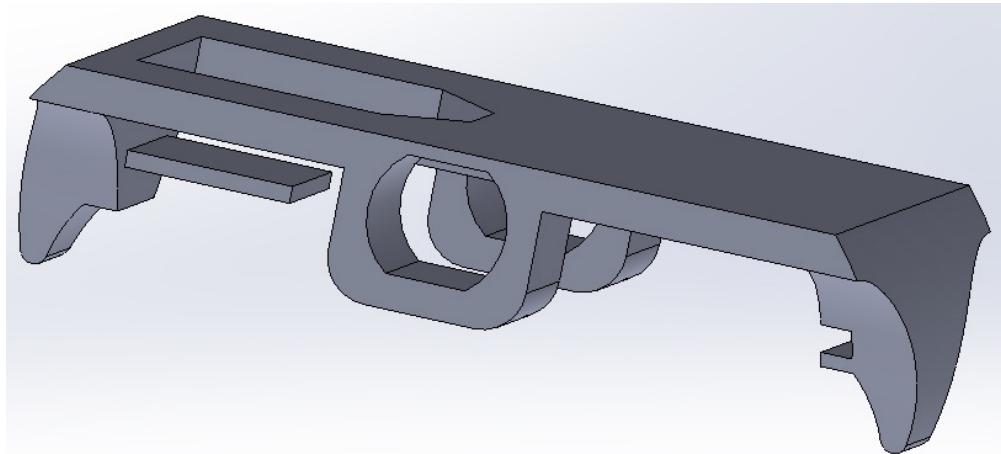


FIGURA 3.9: Primer Chasis de MODI, realizado con SolidWorks 2012. Esta versión sirve solo como prueba de concepto ya que no tiene espacio para las conexiones eléctricas necesarias

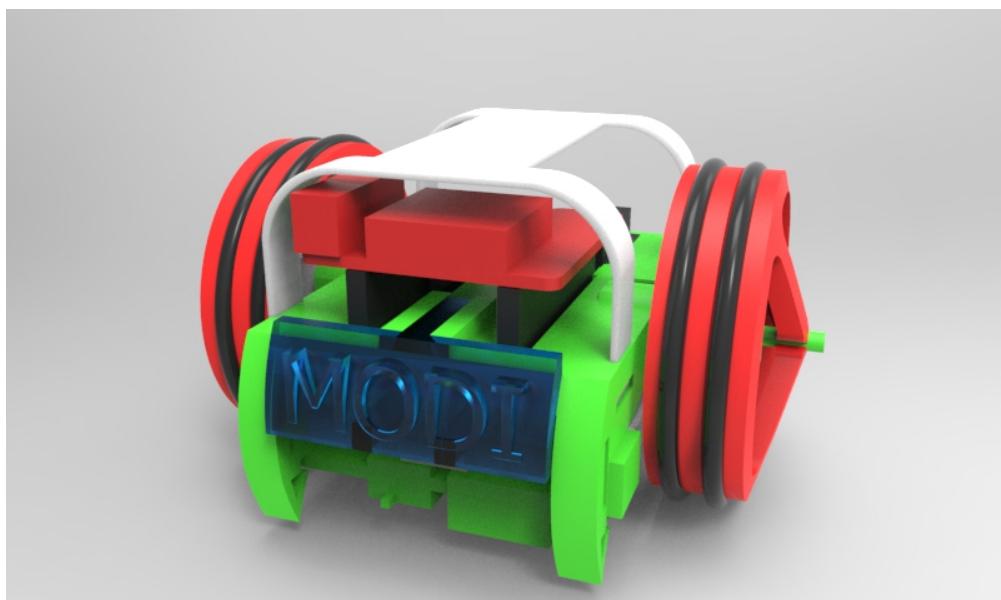


FIGURA 3.10: Modelos 3D de MODI realizado con Autodesk Inventor 2013 y renderizado con KeyShot4.

### 3.2.3. Impresora 3D

Parte importante de este trabajo se realizó con impresoras 3D. Estas existen desde los años 80' y hasta hace algunos años por su precio y tamaño eran de difícil acceso. Utilizan distintas tecnologías, las que se utilizaron en el proceso de este trabajo son las FDM, sigla del inglés "Fused deposition modeling" donde un cabezal extrusor de plástico se mueve en el plano xy depositando capa por capa de material mientras avanza en el eje z.

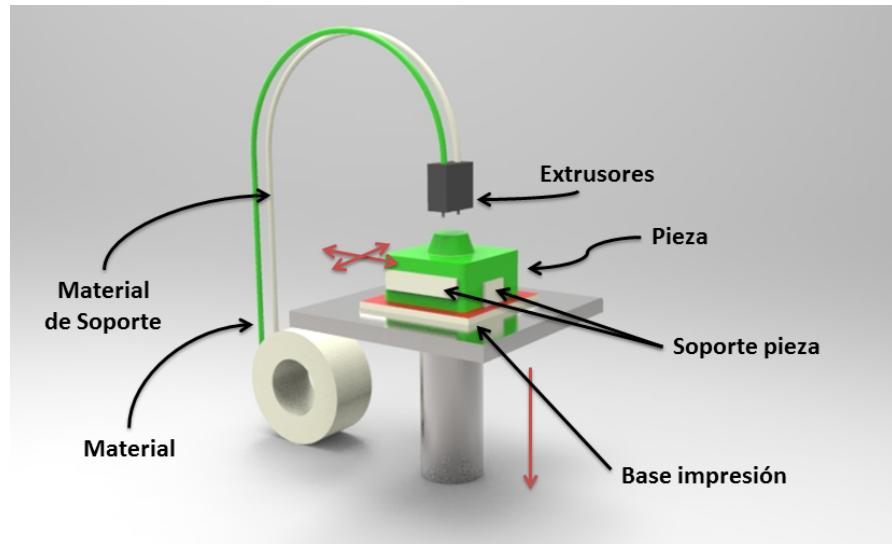


FIGURA 3.11: Modelo de funcionamiento para impresoras 3D MDF. Tiene dos materiales, uno para construir la pieza y otro que funciona como soporte estructural para la impresión, además existe una base que se calienta para sujetar la pieza mientras se construye.

	Replicator 1	3D Touch	Uprint
Volumen impresión	5m <sup>3</sup>	10 m <sup>3</sup>	4.7 m <sup>3</sup>
Espesor capa	0.3 mm	0.125mm	0.254 mm
Velocidad	40 mm/s	6 mm/s	lenta
Precisión	2.5 $\mu$ m Z , 11 $\mu$ m XY	0.1mm Z 0.2mm XY	
Material soporte	con herramienta	con herramienta	Necesita proceso extra
Precio insumo	60USD/Kg	100USD/Kg	400USD/Kg

TABLA 3.1: Comparación impresoras 3D

Primero se utilizó la 3D Touch Dual de la empresa Bit from Bytes, que permite hacer modelos de hasta 185 x 273 x 200[mm], con una buena resolución de 0.125[mm]<sup>1</sup> cada capa en eje z. Luego por facilidad de uso y menor tiempo de impresión se usa una impresora MakerBot Replicator 1

<sup>1</sup>[http://teambastech.com/Store/index.php?route=product/product&product\\_id=205](http://teambastech.com/Store/index.php?route=product/product&product_id=205)



FIGURA 3.12: Impresora 3D de Bit From Bytes, 3D Touch.

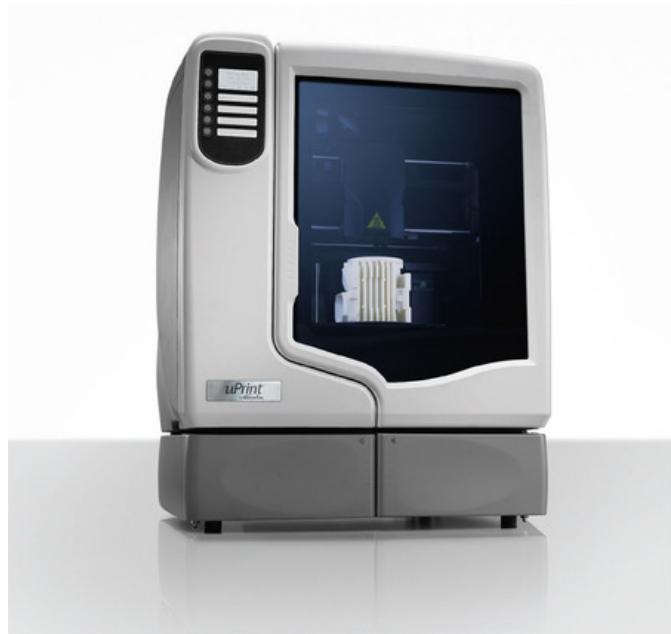


FIGURA 3.13: Impresora 3D de Stratasys, Uprint SE.

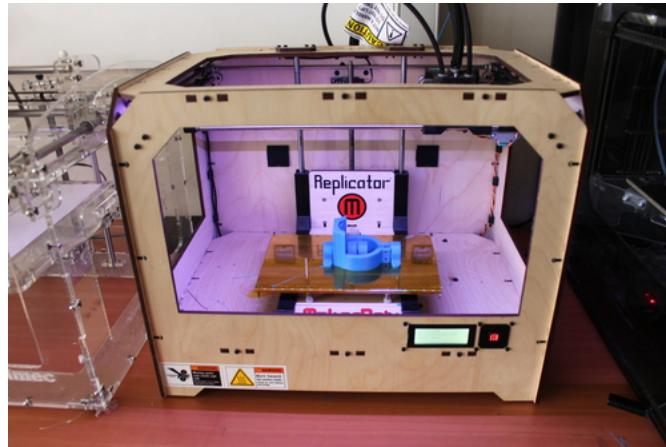


FIGURA 3.14: Replicator 1 de Makerbot, impresora 3D FDM utilizada para prototipar chasis y ruedas de robot MODI. En Laboratorio de Síntesis de maquinas inteligentes de la Universidad de Chile, junto a la Fab@Home y 3DTouch.

### 3.2.4. Diseño PCB

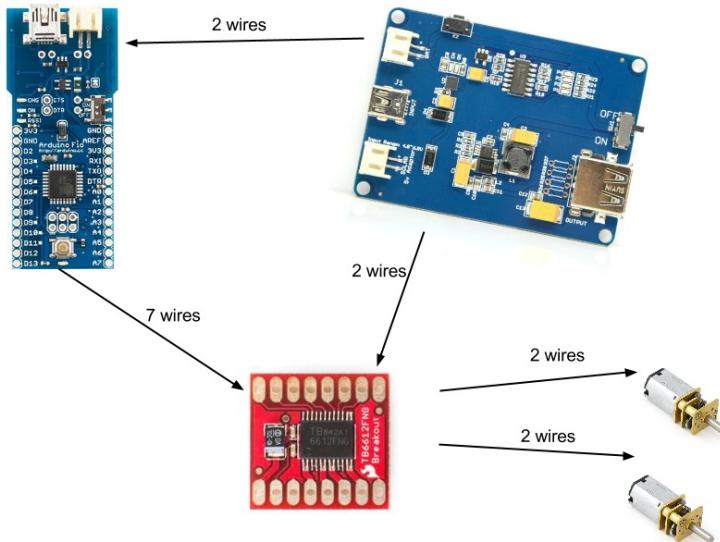


FIGURA 3.15: 15 conexiones que unen distintos componentes.

En los primeros prototipos un problema que se repetía constantemente son las conexiones entre los sistemas que componen al robot. Son 15 conexiones que unen distintos componentes, Figura 3.15. Por esto es necesario un PCB que permita reducir al mínimo el espacio necesario para energizar y comunicar los elementos. El requerimiento principal es poder fabricarse con una fresa CNC como la Roland Modela MDX-20, por

lo que es necesario diseñar un PCB de una sola capa. Se utilizó Eagle como software para el diseño del diagrama eléctrico y PCB, el resultado se puede ver en la Figura 3.17.



FIGURA 3.16: Roland Modelo MDX-20 usada para hacer PCB y modelos 3D. Imagen tomada de [roland.com](http://roland.com)

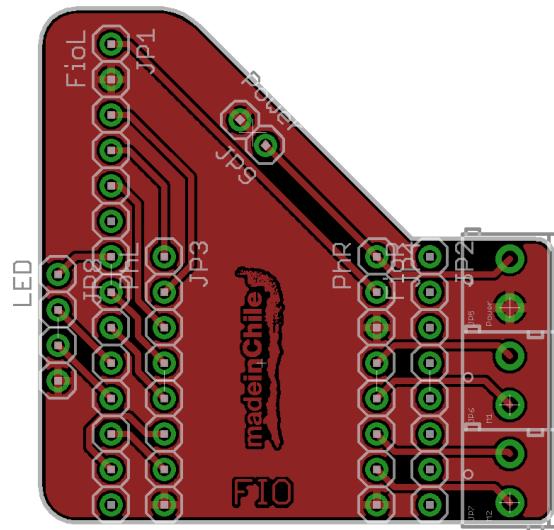


FIGURA 3.17: MODIBoard es una PCB diseñada para poder conectar de forma simple los diversos componentes electrónicos. Conecta: Arduino Fio, Lipo Rider Pro, Motor Driver, LED RGB y Motores.

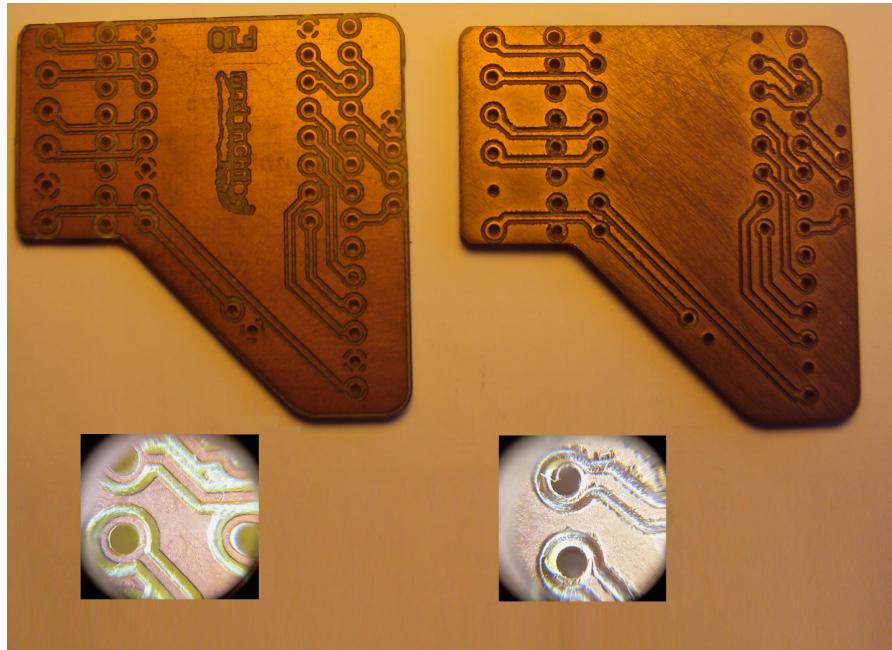


FIGURA 3.18: Misma PCB fresada en dos maquinas diferentes. Se puede apreciar en el detalle, la importancia de contar con una buena resolución de fresado. En especial es necesario para tener un buen pad.

### 3.3. Componentes electrónicos

MODI está compuesto por varios componentes electrónicos que implementan tecnologías nuevas. De forma sencilla el usuario puede hacer uso de todas estas, que por ser Open Source tienen una amplia documentación. Su uso en colegios, talleres y universidades, por ser de bajo costo, y sin partes externas extremadamente frágiles, es una excelente herramienta para iniciar a alumnos en temas como la Programación, Robótica, Comunicación Inalámbrica y Visión Artificial, entre otros, sin los miedos asociados a utilizar un equipo costoso y delicado.

El robot MODI es pequeño y de fácil construcción con las herramientas de Fabricación Digital que se puede encontrar en un FabLab. Tiene un pequeño microcontrolador **Arduino** que controla dos motores, y un **XBee** para la comunicación inalámbrica con un computador central que usando cualquier lenguaje de programación con comunicación serial puede controlar uno o más robots. La posición de cada robot se obtiene con tags **Fiduciales**, que permiten por medio de una cámara cenital saber la coordenada y rotación de cada uno. En esta primera aplicación es necesario contar con autonomía energética por lo que se incluye una batería lipo de 2000[mAh] junto a un módulo que carga la batería y eleva el voltaje de salida a 5[v]. Este el módulo de carga, **Lipo Rider Pro**, tiene además la ventaja de permitir distintos tipos de carga, como una celda de carga solar o un sistema de carga inalámbrica por medio de bobinas. Tener la posibilidad

de cargar una batería con luz solar o por medio de forma inalámbrica hace posible el hacer experimentos que necesiten estar funcionando semanas sin la intervención de un humano para cargar cada uno.

### 3.3.1. Arduino FIO

Microcontrolador El Arduino Fio es una placa para microcontrolador basada en el ATmega328P. Funciona a 3.3V y 8 MHz. Tiene 14 pines de I/O digitales (de los cuales 6 pueden usarse como salidas PWM), 8 entradas analógicas, un oscilador en placa, un botón de reinicio (reset), y agujeros para montar conectores de pines. Tiene conexiones para una batería de polímero de Litio e incluye un circuito de carga a través de USB. En el reverso de la placa tiene disponible un zócalo para módulos XBee.

El Arduino FIO está diseñado para aplicaciones inalámbricas. El usuario puede subir sus sketches con un cable FTDI o una placa adicional adaptadora Sparkfun. Además, si utiliza un adaptador de USB a XBee modificado (como el USB Explorador de XBee), el usuario puede subir sketches de forma inalámbrica. La tarjeta viene sin conectores pre-montados, permitiendo el uso de diversos tipos de conectores o la soldadura directa de los cables. La tabla 3.2 resume sus características principales.

El Arduino FIO ha sido diseñado por Shigeru Kobayashi y SparkFun, y fabricado por SparkFun. Arduino FIO. Similar al arduino mini pro pero además incluye conexión para módulo XBee. Imagen tomada de arduino.cc

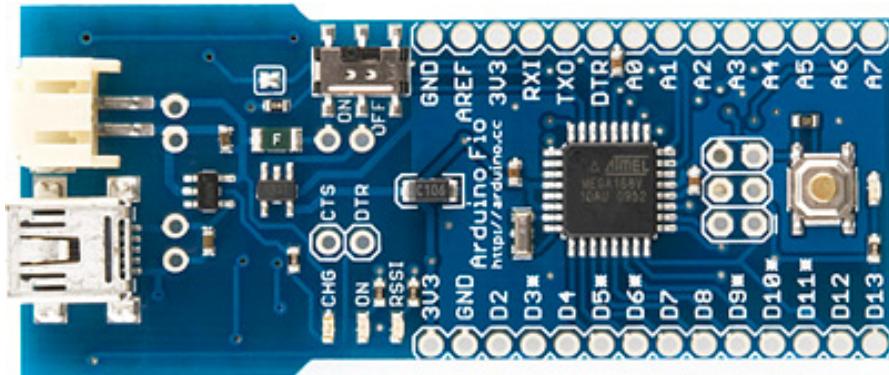


FIGURA 3.19: Arduino FIO. Similar al arduino mini pro pero además incluye conexión para módulo XBee. Imagen tomada de arduino.cc

Microcontrolador	ATmega328P
Voltaje operación	3.3 V
Input Vontaje	3.35 - 12 V
Input Voltaje carga batería	3.7 - 7 V
Pins Digital I/O	14 (con 6 salidas PWM)
Input Pins Analogos	8
Corriente DC en cada Pin I/O	40 mA
Memoria Flash	32 KB (2KB son usados para Bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock	8 MHz

TABLA 3.2: Especificacion tecnica de Arduiono FIO

Velocidad @6 V, 40 mA	440 RPM
Torque	3 Kg-mm
Reducción	1:30
Consumo trancado	0.36 A

TABLA 3.3: Especificación técnica de Motor DC Pololu

### 3.3.2. Motores DC

Este motor Pololu es muy pequeño (largo: 9.27 mm), por lo mismo usa muy poca corriente. Además tiene una caja de reducción metálica de 30:1. El eje tiene forma de D.

Pese a no ser la alternativa más económica, estos motores han sido muy difundidos para su uso en robótica por su bajo consumo y alto torque. La tabla 3.3 resume sus características principales.

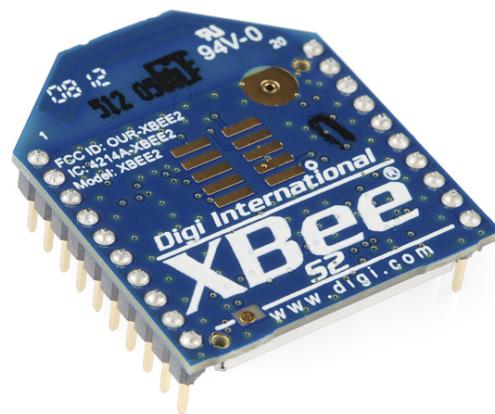
FIGURA 3.20: Motor pololu con reducción 1:30. Imagen tomada de [pololu.com](http://pololu.com)

Velocidad @6 V, 40 mA	440 RPM
Torque	3 Kg-mm
Reducción	1:30
Consumo trancado	0.36 A

TABLA 3.4: Módulo XBee para comunicación inalámbrica. Imagen tomada de [digi.com](http://digi.com)

### 3.3.3. XBee

XBee es un producto de la empresa Digi basado en el conocido protocolo de comunicación inalámbrica ZigBee. El módulo XBee XBP24-AWI-001 funciona a 2.4GHz. La serie Pro tiene el mismo pinout y set de instrucciones que la serie básica más un incremento de la potencia de salida a 60mW. Estos módulos permiten extender la comunicación serial de forma inalámbrica. Con XBee se pueden hacer redes punto a punto y multipunto de forma fácil.

FIGURA 3.21: Motor pololu con reducción 1:30. Imagen tomada de [pololu.com](http://pololu.com)

**3.3.4. Batería****3.3.5. Lipo Rider Pro****3.3.6. Motor Driver 1A Dual TB6612FNG****3.3.7. LED RGB****3.4. Componentes mecánicos****3.5. Locomoción****3.6. Tracking 2D**

# Capítulo 4

## Chapter Title Here

### 4.1. Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

#### 4.1.1. Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

#### 4.1.2. Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget

ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

## 4.2. Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

## **Apéndice A**

### **Appendix Title Here**

Write your Appendix content here.

# Referencias

- [1] Juan Cristóbal Zagal and Hod Lipson. Self-reflection in evolutionary robotics: resilient adaptation with a minimum of physical exploration. In Franz Rothlauf, editor, *GECCO (Companion)*, pages 2179–2188. ACM, 2009. ISBN 978-1-60558-505-5. URL <http://dblp.uni-trier.de/db/conf/gecco/gecco2009c.html#ZagalL09>.
- [2] S. Kernbach, P. Levi, E. Meister, F. Schlachter, and O. Kernbach. Towards self-adaptation of robot organisms with a high developmental plasticity. In *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD '09. Computation World:*, pages 180–187, 2009. doi: 10.1109/ComputationWorld.2009.11.