

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE CIENCIAS  
MAESTRÍA Y DOCTORADO EN CIENCIAS E INGENIERÍA  
MAESTRÍA EN CIENCIAS



---

**LABORATORIO TELEOPERADO DE ROBÓTICA PEDAGÓGICA**

---

TESIS PROFESIONAL  
QUE PARA CUBRIR PARCIALMENTE LOS REQUISITOS PARA  
OBTENER  
EL GRADO DE MAESTRO EN CIENCIAS

PRESENTA:  
TAKAO SANAY ROBLES

DIRECTORES DE TESIS:  
DR. JOSÉ ÁNGEL GONZÁLEZ FRAGA  
DR. OMAR ÁLVAREZ XOCHIHUA

ENSENADA, BAJA CALIFORNIA, MÉXICO, ENERO DEL 2016

Universidad Autónoma de Baja California  
Facultad de Ciencias

Laboratorio Teleoperado de Robótica Pedagógica

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

MAESTRO EN CIENCIAS

Presenta

Takao Sanay Robles

Aprobada por:

---

Dr. José Ángel González Fraga  
Co-Director de Tesis

---

Dr. Omar Álvarez Xochihua  
Co-Director de Tesis

---

Dr. Juan Iván Nieto Hipólito  
Miembro del Comité

---

MC. Evelio Martínez Martínez  
Miembro del Comité

---

MI. Adrián Enciso Almanza  
Miembro del Comité

Ensenada, Baja California, México, Enero del 2016.

**Resumen** de la tesis de **Takao Sanay Robles**, presentada como requisito parcial para la obtención del grado de MAESTRO EN CIENCIAS del programa de Maestría y Doctorado en Ciencias e Ingeniería (MYDCI) de la UABC. Ensenada, Baja California, México, Enero del 2016.

## Laboratorio Teleoperado de Robótica Pedagógica

Resumen aprobado por:



---

Dr. José Ángel González Fraga  
Co-Director de Tesis



---

Dr. Omar Álvarez Xochihua  
Co-Director de Tesis

En este trabajo se presenta el desarrollo y evaluación de un Laboratorio Teleoperado de Robótica Pedagógica (LTRP), el cual tiene como finalidad complementar el proceso de enseñanza-aprendizaje, que se imparte en el aula, de cursos introductorios de programación, principalmente en tópicos de lógica programática. La adquisición de conocimiento se realiza a través de la implementación de prácticas de programación que han sido diseñadas contemplando el modelo instruccional de Dick y Carey, para que el alumno obtenga experiencias de aprendizaje y habilidades de una forma más eficiente, eficaz y atractiva. Se ha desarrollado un lenguaje de programación visual que emplea los símbolos de diagramas de flujo para ayudar a los alumnos en su lógica programática. El LTRP cuenta con una interfaz de usuario amigable que permite al alumno completar los diagramas para después enviarlos al robot, para después observar en tiempo real las acciones de los programas. También permite al profesor generar las prácticas a utilizar en el laboratorio con una interfaz amigable. Resultados preliminares demuestran que los alumnos que utilizaron el LTRP durante el tiempo de estudio de la materia de diseño de algoritmos, lograron reforzar su conocimiento en el tema de estructuras de control. Un aspecto importante que se logró, fue el nivel de motivación que proporciona el LTRP al programar con objetos visuales y trabajar con robots.

**Palabras Clave:** *Laboratorio teleoperado, Robótica pedagógica, Diseño instruccional.*

# Dedicatoria

Quiero dedicar este trabajo:

A mis padres, por haberme forjado como la persona que soy en la actualidad; muchos de mis logros se los debo a ustedes entre los que incluye este. Me formaron con reglas y con algunas libertades, pero al final de cuentas, me motivaron constantemente para alcanzar mis metas.

A mis hermanos, que siempre han estado junto a mí, brindándome su apoyo y confianza en todo lo necesario para cumplir mis objetivos como estudiante.

A mis familiares, porque me han brindado su apoyo incondicional y por compartir conmigo buenos y malos momentos.

# Agradecimientos

Quiero expresar mi agradecimiento:

A mis sinodales Dr. José Ángel González Fraga, Dr. Omar Álvarez Xochihua, Dr. Juan Iván Hipólito Nieto, MC. Evelio Martínez Martínez y MI. Adrián Enciso Almanza por sus consejos, críticas y apoyo durante el desarrollo de esta tesis.

A mis compañeros y amigos, que juntos hemos caminado, tropezado y levantado en esta etapa, por los buenos momentos que pasamos y el apoyo que brindaron para concluir mis estudios.

A mi novia, por sus palabras y confianza, por su amor y brindarme el tiempo necesario para realizarme profesionalmente.

Al personal administrativo de la Facultad de Ciencias y el personal del programa de Maestría y Doctorado en Ciencias e Ingeniería (MyDCI) por su atención, amabilidad y disposición para ayudarme.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico brindado y a la Facultad de Ciencias de la Universidad Autónoma de Baja California (UABC) por las facilidades otorgadas para la realización de éste trabajo.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del problema . . . . .	3
1.1.1. Justificación . . . . .	5
1.1.2. Objetivos de la investigación . . . . .	6
1.1.2.1. Objetivo general . . . . .	6
1.1.2.2. Objetivos específicos . . . . .	6
1.1.3. Preguntas de investigación . . . . .	6
1.1.4. Viabilidad de la investigación . . . . .	6
1.2. Estructura de la tesis . . . . .	7
<b>2. Marco teórico</b>	<b>8</b>
2.1. Diseño instruccional . . . . .	8
2.1.1. Modelos de diseño instruccional . . . . .	9
2.1.1.1. Modelo de Gagné . . . . .	9
2.1.1.2. Modelo ASSURE . . . . .	10
2.1.1.3. Modelo genérico (ADDIE) . . . . .	11
2.1.1.4. Modelo de Dick y Carey . . . . .	11
2.2. Estrategias para la enseñanza . . . . .	13
2.2.1. Estilos de aprendizaje . . . . .	14
2.2.2. Teoría construccionista . . . . .	14
2.2.3. Enseñanza de la programación . . . . .	15
2.3. Tecnologías de la información como apoyo en la educación . . . . .	16
2.3.1. Robótica pedagógica . . . . .	16
2.3.1.1. Juego de robótica educativa . . . . .	18
2.4. Laboratorios virtuales y teleoperados . . . . .	19
2.4.1. Laboratorios virtuales y teleoperados de robótica . . . . .	19
<b>3. Diseño de prácticas instruccionales</b>	<b>22</b>
3.1. Selección de temáticas . . . . .	22
3.1.1. Identificación de temáticas objetivo . . . . .	23
3.1.2. Materiales . . . . .	23
3.1.3. Selección de participantes . . . . .	23
3.1.4. Temáticas factibles de implementación . . . . .	23

3.1.5. Implementación de ejercicios . . . . .	25
3.2. Actividad 1: “Simulando un robot” . . . . .	27
3.2.1. Objetivo . . . . .	31
3.2.2. Resultados . . . . .	31
3.2.3. Conclusión . . . . .	33
3.3. Actividad 2: Diagramas de flujo y programación con Scratch . . . . .	33
3.3.1. Objetivo . . . . .	43
3.3.2. Resultados . . . . .	43
3.3.3. Conclusión . . . . .	44
3.4. Actividad 3: Programación en un ambiente web . . . . .	44
3.4.1. Objetivo . . . . .	45
3.4.2. Resultados . . . . .	45
3.4.3. Conclusión . . . . .	47
<b>4. Diseño e implementación del LTRP</b>	<b>48</b>
4.1. Componentes de Hardware . . . . .	49
4.1.1. Infraestructura . . . . .	49
4.1.2. Conexión de cámaras . . . . .	50
4.1.3. Armado del robot . . . . .	51
4.1.4. Evidencia fotográfica . . . . .	52
4.2. Componentes de Software . . . . .	53
4.2.1. Ambiente de programación . . . . .	54
4.2.1.1. Ambiente del maestro . . . . .	54
4.2.1.2. Ambiente del alumno . . . . .	58
4.2.1.3. Programa del robot . . . . .	62
<b>5. Evaluación del LTRP</b>	<b>63</b>
5.1. Métodos utilizados en el estudio . . . . .	63
5.2. Participantes . . . . .	63
5.3. Duración del estudio . . . . .	64
5.4. Fuente de datos y mecanismos de análisis . . . . .	66
5.5. Resultados del estudio . . . . .	66
5.5.1. Beneficios obtenidos en el nivel de aprovechamiento de los alumnos	66
5.5.2. Aceptación en el uso del ambiente de aprendizaje . . . . .	67
5.5.3. Análisis de vídeos . . . . .	68
5.5.3.1. Ambiente de programación . . . . .	68
5.5.3.2. Visualización del robot . . . . .	69
<b>6. Conclusiones generales</b>	<b>71</b>
6.1. Conclusiones . . . . .	71
6.2. Aportaciones . . . . .	73
6.3. Trabajo a futuro . . . . .	73

<b>A. Encuesta</b>	<b>74</b>
A.1. Encuesta factores principales que inhiben el aprendizaje de programación	74
<b>B. Guía para diseñar cursos</b>	<b>79</b>
B.1. Introducción . . . . .	79
B.2. Identificar metas instruccionales . . . . .	80
B.3. Análisis instruccional . . . . .	80
B.4. Objetivos y redacción de instrumentos de evaluación . . . . .	82
B.5. Estrategia instruccional . . . . .	83
B.6. Materiales instruccionales . . . . .	84
B.7. Evaluación formativa . . . . .	85
B.8. Revisión de instrucción . . . . .	85
B.9. Conclusión . . . . .	86
<b>C. Pre-test y Post-test</b>	<b>87</b>

# Índice de figuras

1.1. Total de estudiantes en carreras de computación en UABC campus Ensenada. . . . .	4
1.2. Total de estudiantes en carreras de computación en UABC campus Tijuana. . . . .	4
1.3. Total de estudiantes en carreras de computación en UABC campus Mexicali. . . . .	5
2.1. Modelo ASSURE. . . . .	10
2.2. Modelo de Dick y Carey. Fuente: <a href="http://proesa.wikispaces.com/">http://proesa.wikispaces.com/</a> . . . . .	12
2.3. Ejemplos de robots pedagógicos. . . . .	18
3.1. Respuestas de la pregunta. ¿Qué se les dificulta a los alumnos? . . . . .	24
3.2. Respuestas de la pregunta. ¿Cuándo crees que aprendes más? Sesiones de prácticas, estudiar por tu cuenta. . . . .	24
3.3. Respuestas de la pregunta. ¿Qué material crees que te ayuda para aprender a programar? Material interactivo, programas de ejemplo, ejercicios de preguntas y respuestas. . . . .	25
3.4. Ejecutando el algoritmo I. . . . .	28
3.5. Ejecutando el algoritmo II. . . . .	28
3.6. Palabras a utilizar. . . . .	29
3.7. Pizarrón de prácticas. . . . .	29
3.8. Ejemplo de práctica. . . . .	30
3.9. Diagrama ruta del robot. . . . .	31
3.10. Práctica 1 primera parte. Nivel de dificultad bajo. . . . .	35
3.11. Práctica 1 segunda parte. Nivel de dificultad alto. . . . .	36
3.12. Práctica 2. Nivel de dificultad alto. . . . .	37
3.13. Práctica 1 primera parte. Respuestas. . . . .	38
3.14. Práctica 1 segunda parte. Respuestas. . . . .	39
3.15. Práctica 2. Respuestas. . . . .	40
3.16. Práctica “Guaca Gobo”. . . . .	41
3.17. Código pong en Scratch. . . . .	42
3.18. Código de la raqueta en Scratch. . . . .	43
3.19. Resultados XXXI Semana de Ciencias. . . . .	44

3.20. Ambiente de trabajo primera versión. Interfaz del alumno. . . . .	45
3.21. Resultados experimento COBACH. . . . .	46
4.1. Arquitectura propuesta. . . . .	49
4.2. Conexión de cámaras. . . . .	51
4.3. Mundo del robot. . . . .	52
4.4. Laboratorio Teleoperado de Robótica Pedagógica. . . . .	53
4.5. Menú principal. . . . .	54
4.6. Opciones de movimiento. . . . .	55
4.7. Opciones de estructuras de control. . . . .	55
4.8. Ejemplo como ir creando un ejercicio. . . . .	56
4.9. Ejemplo del archivo XML correspondiente al ejercicio de la figura 4.8.	56
4.10. Flujo para crear práctica en XML. . . . .	57
4.11. Diagrama: Conversión HTML - XML. . . . .	57
4.12. Ambiente de programación segunda versión. Interfaz del maestro . . .	58
4.13. Opciones de sensores. . . . .	58
4.14. Ejemplo de ejercicio para el alumno. . . . .	59
4.15. Flujo para cargar el archivo en XML en el ambiente. . . . .	59
4.16. Diagrama: Conversión XML - HTML. . . . .	60
4.17. Inicio de la práctica. . . . .	60
4.18. Práctica compilada y pseudocódigo. . . . .	61
4.19. Visualización del robot I. . . . .	61
4.20. Visualización del robot II. . . . .	62
5.1. Ejemplo de práctica del LTRP. . . . .	65
5.2. Ambiente de programación I. . . . .	69
5.3. Ambiente de programación II. . . . .	69
5.4. Visualización del robot. . . . .	70
A.1. * . . . .	75
A.2. * . . . .	76
A.3. * . . . .	77
A.4. * . . . .	78
B.1. Proceso para conseguir meta. . . . .	81
B.2. Proceso para conseguir meta dividida en actividades. . . . .	81
C.1. Examen pre-test y post-test I. . . . .	88
C.2. Examen pre-test y post-test II. . . . .	89
C.3. Examen pre-test y post-test III. . . . .	90
C.4. Examen pre-test y post-test IV. . . . .	91

# Índice de tablas

3.1. Encuesta aplicada a los participantes de la actividad Simulando un robot	32
5.1. Relación preguntas-competencia . . . . .	64
5.2. Resultados Pre-test y Post-test. . . . .	67
5.3. Comentarios de los entrevistados en categorías. . . . .	68
B.1. Ejemplo de la etapa de identificación de metas. . . . .	80
B.2. Objetivos e instrumentos de evaluación por actividad. . . . .	83
B.3. Diseñar evaluaciones PRE y POST. . . . .	84
B.4. Actividades divididas en sesiones. . . . .	84
B.5. Material instruccional. . . . .	85
B.6. Revisión de materiales. . . . .	85

# Capítulo 1

## Introducción

La palabra robot fue utilizada por primera vez en 1921 por el escritor checo Karel Capek en su obra “Rossum’s Universal Robot (R.U.R.)”. Su origen viene de la palabra checa “robota”, que se refiere a: “trabajo realizado de manera forzada”. En 1948, George Devol, considerado como pionero de la robótica industrial, patentó un manipulador programable que fue el germen de la robótica industrial [1]. El científico y escritor de ciencia ficción Issac Asimov, utilizó la palabra robótica por primera vez y la definió como “una ciencia o rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano o que requieren del uso de inteligencia” [1]. Con el pasar de los años, se han creado máquinas para realizar diferentes actividades, desde tareas simples y de esparcimiento, hasta tareas peligrosas al ser humano, como exploraciones marinas, volcánicas y misiones a Marte.

Con base en el paradigma construccionista, ampliamente utilizado en el entorno educativo, tenemos que la robótica y las computadoras en general, juegan un rol importante en la formación de estudiantes desde edades tempranas [2]. El construccionismo es una teoría de aprendizaje desarrollada por Seymour Papert, basada en la ideología del constructivismo desarrollada por Jean Piaget [3]. Esta teoría parte del supuesto que el conocimiento se construye dependiendo de la acción que va realizando el estudiante y le provoque emoción o satisfacción, por ejemplo, al construir un castillo de arena o al construir un robot con piezas lego, el estudiante aprende, entre otras cosas, las propiedades de los materiales.

Derivado de las ventajas que se presentan al trabajar con robots y la teoría del construccionismo, se definió la disciplina de robótica pedagógica, que permite diseñar y desarrollar robots educativos para que los estudiantes se inicien en el estudio de la ciencia y la tecnología. También, tiene el alcance de proporcionar instrucción en otras áreas de conocimiento, como por ejemplo las matemáticas, ciencias naturales, entre otras [4].

Por medio de la robótica pedagógica, se privilegia el aprendizaje inductivo y por descubrimiento guiado. En el aprendizaje inductivo el estudiante trata de aprender a partir

de datos, analizando la información y comparando, con esto, puede construir sus propios conceptos o hipótesis. A su vez, en el aprendizaje por descubrimiento guiado, el docente se convierte en guía del estudiante. El estudiante juega el papel del descubridor y participa en las actividades de planificar, programar, ejecutar y evaluar el proceso de aprendizaje [5]. La inducción y el descubrimiento guiado aseguran que los estudiantes diseñen y experimenten las situaciones didácticas que permitirán construir su propio conocimiento [6].

Así mismo, las nuevas generaciones de estudiantes emplean ampliamente las Tecnologías de Información y Comunicación (TIC) en su vida cotidiana, esto debido a la gran disponibilidad y accesibilidad de computadoras de última generación, teléfonos inteligentes, Internet y demás. Actualmente, se aprovechan estas tecnologías para que los estudiantes refuercen y practiquen conocimientos teóricos adquiridos en las aulas o realizando prácticas en laboratorios equipados con computadoras y otros dispositivos electrónicos orientados a reforzar el aprendizaje. Sin embargo, no todas las instituciones educativas cuentan con la capacidad económica, personal capacitado, ni el espacio para implementar laboratorios con tecnología de punta, por lo que una de las alternativas para solucionar este problema es desarrollar laboratorios virtuales o teleoperados [7]. Los laboratorios virtuales son simulaciones en computadora que utilizan modelos matemáticos para representar el comportamiento o funcionamiento de un sistema que existe en la realidad. Mientras que, los laboratorios teleoperados, son ambientes donde se le permite al estudiante manipular remotamente dispositivos físicos; generalmente a través de Internet. Existen diferentes tipos de estos laboratorios dependiendo del área de conocimiento que se quiera atender.

Los laboratorios virtuales/teleoperados de robótica han sido principalmente utilizados como un complemento en la enseñanza de la manipulación de robots y sus sensores [8, 9, 10, 11, 12]. Estos laboratorios brindan acceso al control de robots o la simulación de los mismos. Con esto, los alumnos pueden llevar a la práctica los conocimientos teóricos de robótica adquiridos en el aula. Sin embargo, este tipo de laboratorios, se enfocan más en la configuración y manipulación de robots a través de comandos básicos. Esto lo podemos considerar como un proceso no estructurado de programación, ya que la manipulación se basa en un conjunto de comandos o instrucciones predefinidas, que limitan la aplicación de la lógica programática en forma integral por parte del estudiante. A diferencia de los objetivos que persiguen los laboratorios de robótica que se han citado, en este trabajo de investigación, se desarrolló y evaluó un laboratorio teleoperado de robótica que complementa la enseñanza de lógica programática a través de la elaboración de prácticas de programación, diseñadas con un modelo instruccional, que permitieron fortalecer aquellos conceptos que se dificultan en esta área de estudio y en donde los resultados del programa dan acceso para manipular a un robot móvil.

## 1.1. Planteamiento del problema

Un mundo globalizado y el constante avance tecnológico, generan un ambiente competitivo donde empresas e instituciones que deseen permanecer y evolucionar en este entorno deben adoptar una cultura de actualización permanente en relación a Tecnologías de Información y Comunicación. Lo anterior, ha generado un incremento significativo en la demanda de profesionistas en TIC, siendo especialistas en redes, seguridad, desarrollo de software, entre otros, altamente solicitados en el entorno laboral [13]. Específicamente, en el área de desarrollo de software, La Conferencia de las Naciones Unidas sobre Comercio y Desarrollo (UNCTAD, por sus siglas en inglés), reportó un incremento del 40 % al 80 % en la demanda de profesionistas en este sector [14].

En contraste, a nivel internacional, se presenta un fenómeno que impacta negativamente en la formación de profesionistas que cubran este requerimiento laboral. Existe una reducción en el número de jóvenes interesados en estudiar programas profesionales orientados a las TIC [15]. Adicionalmente, el índice de estudiantes que desertan de estas carreras es igualmente preocupante. Siendo estas dos situaciones, escaso ingreso y el alto índice de deserción escolar, factores que detonan una insuficiencia en la cantidad de profesionistas que atiendan la demanda actual.

Haciendo referencia al problema de deserción escolar, las instituciones de educación superior, tanto nacionales como internacionales, enfrentan niveles de baja académica de entre el 30 % y 50 % [16]. Por ejemplo, [16] reportan que aproximadamente el 65 % no aprueban los cursos introductorios de Ciencias Computacionales a nivel mundial (Estados Árabes, Centro y Este de Europa, Asia Central, Latinoamérica y El Caribe, Sur y Oeste de Asia y Europa Occidental). En el caso particular de México, la coordinación de la Red de Tutorías de la Región Centro-Occidente, de la Asociación Nacional de Universidades e Instituciones de Educación Superior (ANUIES), informó sobre el comportamiento nacional de este fenómeno, indicando que alrededor del 25 % de los estudiantes universitarios abandonan su carrera en el primer semestre, alcanzando un 36 % en el tercer semestre. Agravando el problema en las áreas de Ciencias Naturales y Exactas, e Ingeniería y Tecnología [17].

Esta situación también es evidente en las diferentes licenciaturas con perfil en computación que oferta la Universidad Autónoma de Baja California. Un comportamiento decreciente de la matrícula en estas carreras desde 2008 se puede apreciar en las figuras 1.1, 1.2 y 1.3 (Fuente: Elaboración propia con datos obtenidos de:

<http://www.uabc.mx/planeacion/sii/Alumnos/Absoluto/index.html>).

Siendo matemáticas y programación los cursos con mayor índice de reprobación y detonantes de deserción. Específicamente, en cursos de programación, área de interés de nuestro trabajo de investigación, profesores en nuestra institución reportan que el índice de reprobación alcanza hasta un 40 %.

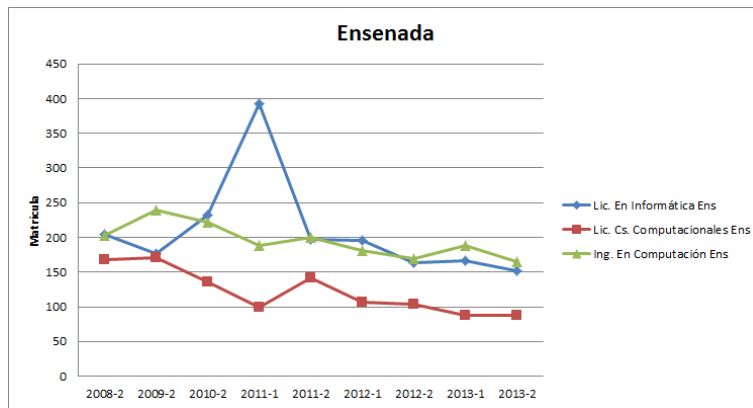


Figura 1.1: Total de estudiantes en carreras de computación en UABC campus Ensenada.

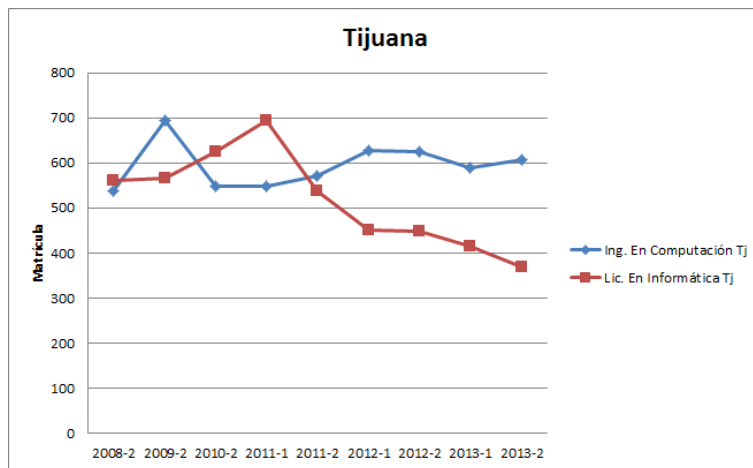


Figura 1.2: Total de estudiantes en carreras de computación en UABC campus Tijuana.

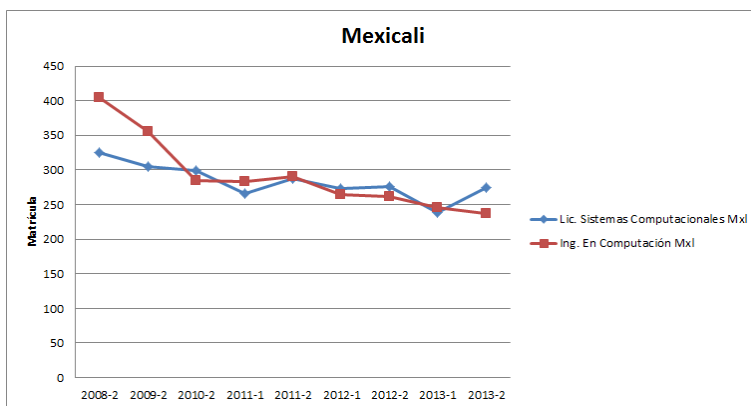


Figura 1.3: Total de estudiantes en carreras de computación en UABC campus Mexicali.

Investigaciones empíricas reportan que dentro de las principales causas que impiden la obtención de competencias satisfactorias en el área de programación se encuentran: 1) la ausencia de ejercicios que despierten el interés del estudiante [15, 18], 2) pocas horas de práctica de laboratorio [18], y 3) dificultad para entender y aplicar conceptos como abstracción de datos, modularidad, recursividad, estructuras de control, apuntadores y depuración de errores [15, 19, 20].

Ante esta situación ambivalente, nos motiva el definir mecanismos para 1) mantener motivados a los estudiantes actuales, 2) proporcionar un entorno instruccional que facilite lograr las competencias requeridas en programación y 3) apoyar a la generación de recursos humanos competitivos y de alta calidad para incorporarse al sector laboral.

En la presente investigación se propuso desarrollar un ambiente que facilite la enseñanza de lógica programática utilizando robots pedagógicos; como herramienta de apoyo en el aprendizaje de programación de computadoras. Buscando atender las tres principales dificultades en la enseñanza de programación expuestas en la literatura: 1) la ausencia de ejercicios que despierten el interés del estudiante; 2) pocas horas de laboratorio-práctica; 3) dificultad para entender y aplicar conceptos como abstracción de datos, modularidad, recursividad, estructuras de control, apuntadores y depuración de errores. De esta manera se busca apoyar en la reducción del índice de reprobación de estudiantes de cursos de programación e impactar en el incremento de profesionistas en el área de las TIC.

### 1.1.1. Justificación

La realización de esta investigación tiene dos motivos principales que la justifican. Primero, una aportación de carácter tecnológico al área de enseñanza de las ciencias computacionales, con el desarrollo de un laboratorio de robótica pedagógica teleoperado, que influya positivamente en la formación de estudiantes en el área de programación,

a nuestro conocimiento, no disponible actualmente. Segundo, el contar con una metodología para el diseño y elaboración de las prácticas interactivas en el LTRP, que permita a profesores del área contar con una guía validada para la creación de sus propios ejercicios en el laboratorio propuesto o ambientes educativos similares.

## 1.1.2. Objetivos de la investigación

### 1.1.2.1. Objetivo general

- Implementar un Laboratorio Teleoperado de Robótica Pedagógica (LTRP) que utilice Diseño Instruccional para desarrollar prácticas interactivas de programación, que permita la adquisición de competencias en el área.

### 1.1.2.2. Objetivos específicos

- Identificar las temáticas de programación, con mayor problema de comprensión y aplicación (ej. abstracción de datos, modularidad, recursividad, estructuras de control, apuntadores y depuración de errores), que son factibles de implementar a través de prácticas interactivas en el LTRP.
- Identificar las características (ej. complejidad, tiempo de instrucción, tipo de interacción y nivel de colaboración) de las prácticas interactivas en el LTRP, que permitan incrementar el interés del estudiante y reforzar el aprendizaje de las temáticas de programación.
- Definir una metodología para la elaboración de las prácticas interactivas en el LTRP, con base en la teoría de diseño instruccional.

## 1.1.3. Preguntas de investigación

- ¿Cuáles son las temáticas de programación que pueden ser implementadas a través de prácticas interactivas en el LTRP?
- ¿Cuáles son las características que deben ser consideradas al elaborar las prácticas interactivas, que mantengan el interés del alumno y refuercen el aprendizaje de los temas?
- ¿Qué elementos de las metodologías de diseño instruccional deben ser considerados para la elaboración de las prácticas interactivas en el LTRP?

## 1.1.4. Viabilidad de la investigación

La viabilidad del estudio estuvo basada en la disponibilidad de; (1) infraestructura necesaria para la elaboración del LTRP; (2) colaboradores con experiencia en el uso de metodologías de diseño instruccional para el desarrollo de los ejercicios; y (3) grupos

de estudiantes en cursos del área de programación que pueden ser considerados en el diseño y la evaluación del desempeño y eficiencia del LTRP.

## 1.2. Estructura de la tesis

Este trabajo está dividido en los siguientes capítulos:

**Capítulo 1** En éste capítulo se presenta una introducción del trabajo de tesis, se indica el planteamiento del problema, la justificación, los objetivos tanto general como específicos, preguntas de investigación, viabilidad de la investigación y la secuencia del trabajo.

**Capítulo 2** En el capítulo 2 se presenta el marco teórico, incluyendo los antecedentes, definiciones y descripción detallada de las ventajas de los laboratorios teleoperados y de la robótica pedagógica.

**Capítulo 3** En este capítulo se presenta el diseño de las prácticas utilizadas en las tres actividades (“Simulando un robot”, “Ambiente en papel y programación con Scratch”, “Ambiente web y programación con Scratch”) que se desarrollaron basándonos en diseño instruccional, también en este capítulo se muestran los resultados y conclusiones de cada una de las actividades.

**Capítulo 4** En el capítulo 4 se describe el diseño e implementación del laboratorio desarrollado, se incluye los dispositivos que se utilizaron, la construcción del robot y el framework de programación que se implementó.

**Capítulo 5** En este capítulo se describe el diseño de la evaluación del LTRP y los resultados obtenidos.

**Capítulo 6** Finalmente, en este capítulo se presentan las conclusiones del trabajo.

# Capítulo 2

## Marco teórico

En este capítulo se describe el marco teórico del trabajo de tesis donde podremos observar lo que es el diseño instruccional y algunos modelos de diseño instruccional, como por ejemplo, el modelo ASSURE o el modelo Dick & Carey. Así mismo, se explicarán el estudio de algunas estrategias y teorías de aprendizaje dentro y fuera de las ciencias computacionales, además se discutirá como la tecnología ha ayudado en la educación para crear herramientas de apoyo en el proceso de la enseñanza-aprendizaje. Por último, se discutirán algunos de los trabajos más relevantes relacionados con laboratorios virtuales y teleoperados de robótica.

### 2.1. Diseño instruccional

Tradicionalmente, en el proceso de enseñanza-aprendizaje, los actores principales han sido maestros, alumnos y libros de textos. Donde el conocimiento a transmitir proviene directamente de la experiencia de los docentes y materiales impresos o electrónicos. Siendo responsabilidad del maestro, el enseñar apropiadamente el conocimiento a los alumnos. Lo anterior, implica que el aprendizaje depende directamente del conocimiento de los maestros y los métodos de enseñanza utilizados [21].

Sin embargo, aun cuando el profesor sigue siendo un elemento fundamental en el proceso de enseñanza-aprendizaje, los avances pedagógicos y tecnológicos nos han llevado a tener estudiantes cada vez más autodidactas; implicando la necesidad de contar con contenidos instruccionales efectivos y una mínima o nula dependencia de un instructor.

El diseño instruccional es un proceso sistemático para crear un ambiente de enseñanza y materiales didácticos, consistente de actividades novedosas, para fomentar que los alumnos aprendan mejor y más rápido. Siendo su objetivo principal, el estructurar el proceso de enseñanza-aprendizaje con un enfoque basado en el estudiante [21, 22].

Existen diferentes modelos para el diseño instruccional, basados en diversas teorías

de aprendizaje e instrucción. Las de mayor relevancia son descritas a continuación [22]:

- **Aprendizaje por descubrimiento.** En el aprendizaje por descubrimiento el docente se convierte en guía del estudiante. El estudiante juega el papel del descubridor y participa en las actividades de planificar, programar, ejecutar y evaluar el proceso de aprendizaje.
- **Constructivismo.** El conocimiento no es una copia de la realidad, sino una construcción del ser humano. Esta construcción se realiza con los esquemas que la persona ya posee (conocimientos previos), es decir, con lo que ya construyó en su relación previa con el medio que lo rodea. El constructivismo tiene las siguientes consecuencias para la formación de un sujeto: (1) el alumno construye su propia comprensión, (2) el nuevo aprendizaje depende de la comprensión actual, y (3) el aprendizaje se facilita por la interacción social.

### 2.1.1. Modelos de diseño instruccional

Existen numerosos modelos de diseño instruccional. Estos modelos intentan describir el proceso por el cual se produce una formación de calidad.

#### 2.1.1.1. Modelo de Gagné

El autor sistematiza un enfoque integrador donde se consideran aspectos de las teorías de estímulos-respuesta y de modelos de procesamiento de información. Gagné considera que deben cumplirse, al menos, diez funciones en la enseñanza para que tenga lugar un verdadero aprendizaje [23].

1. Estimular la atención y motivar.
2. Dar información sobre los resultados esperados.
3. Estimular el recuerdo de los conocimientos y habilidades previas, esenciales y relevantes.
4. Presentar el material a aprender.
5. Guiar y estructurar el trabajo del aprendiz.
6. Provocar la respuesta.
7. Proporcionar retroalimentación.
8. Promover la generalización del aprendizaje.
9. Facilitar el recuerdo.
10. Evaluar la realización.

### 2.1.1.2. Modelo ASSURE

Heinich et al. [23] desarrollaron el modelo ASSURE incorporando los eventos de instrucción de Robert Gagné para asegurar el uso efectivo de los medios en la instrucción. El modelo ASSURE tiene sus raíces teóricas en el constructivismo, partiendo de las características concretas del estudiante, sus estilos de aprendizaje y fomentando la participación activa y comprometida del estudiante. ASSURE presenta seis fases o procedimientos los cuales se muestran en la figura 2.1.

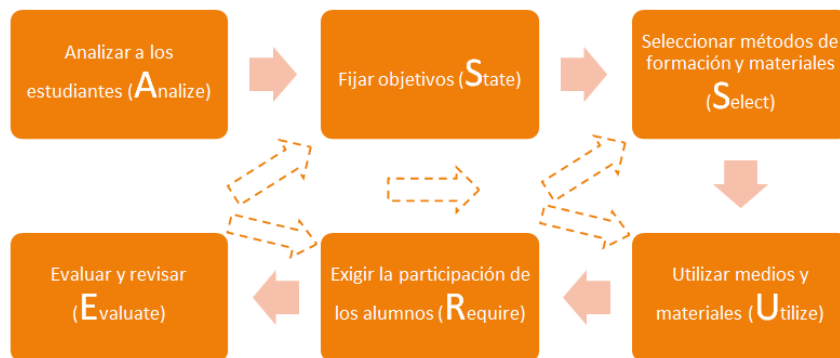


Figura 2.1: Modelo ASSURE.

1. Analizar las características del estudiante. Antes de comenzar, se deben conocer las características de los estudiantes, en relación a:
  - Características generales, como el nivel de estudios, edad, características sociales, físicas, etc.
  - Capacidades específicas de entrada, tales como conocimientos previos, habilidades y actitudes.
  - Estilos de Aprendizaje.
2. Establecimiento de objetivos de aprendizaje, determinando los resultados que los estudiantes deben alcanzar al realizar el curso, indicando el grado en que serán conseguidos.
3. Selección de estrategias, tecnologías, medios y materiales.
  - Método instruccional que se considera más apropiado para lograr los objetivos para esos estudiantes particulares.
  - Los medios que serían más adecuados pueden ser texto, imágenes, video, audio, y multimedia.
  - Los materiales que servirán de apoyo a los estudiantes para el logro de los objetivos.

4. Organizar el escenario de aprendizaje. Desarrollar el curso creando un escenario que propicie el aprendizaje, utilizando los medios y materiales seleccionados anteriormente. Revisión del curso antes de su implementación, especialmente si se utiliza un entorno virtual comprobar el funcionamiento óptimo de los recursos y materiales del curso.
5. Participación de los estudiantes. Fomentar a través de estrategias activas y cooperativas la participación del estudiante.
6. Evaluación y revisión de la implementación y resultados del aprendizaje. La evaluación del propio proceso llevará a la reflexión sobre el mismo y a la implementación de mejoras que redunden en una mayor calidad de la acción formativa.

#### 2.1.1.3. Modelo genérico (ADDIE)

Este modelo tiene 5 etapas de las cuales se deriva su nombre: Análisis, Diseño, Desarrollo, Implementación y Evaluación.

- La etapa de *análisis* contempla la habilidad, actitud y motivación del alumno, así como su perfil cultural. Revisa el contexto de aprendizaje y también define el propósito de la instrucción.
- La etapa de *diseño* describe los objetivos de la unidad, diseña la evaluación, escoge los medios y el sistema para hacer llegar la información, determina el enfoque didáctico general, decide las partes y el orden del contenido, diseña las actividades del alumno e identifica los recursos.
- La tercera etapa, *desarrollo*, identifica las metas del curso, selecciona y se desarrolla contenido didáctico, implementa estrategias instruccionales y define esquemas de evaluación.
- La etapa de *implementación*, debe retroalimentar al estudiante y actualizar el diseño inicial.
- En la última etapa, *evaluación*, se realiza una evaluación formativa y sumativa, interpreta los resultados de las evaluaciones, revisa las actividades, realiza ajustes al modelo. La evaluación se realiza durante todo el proceso y no al final.

#### 2.1.1.4. Modelo de Dick y Carey

Walter Dick y Lou Carey desarrollaron un modelo para el diseño de contenidos o ambientes instruccionales basado en la idea de que existe una relación predecible y fiable entre un estímulo (materiales didácticos) y la respuesta que se produce en un alumno (el aprendizaje de los materiales). El diseñador tiene que identificar las competencias y habilidades que el alumno debe dominar y a continuación seleccionar el estímulo y la estrategia instruccional para su presentación.

Este modelo se basa en los tres enfoques principales del aprendizaje:

1. Aprendizaje como fortalecimiento a la respuesta.
2. Aprendizaje como adquisición de conocimiento.
3. Aprendizaje como construcción del conocimiento.

El primero, es un enfoque conductista del aprendizaje, el alumno recibe pasivamente recompensas y castigos (estímulo-respuesta). El segundo es un enfoque cognitivo basado en que el aprendizaje tiene lugar cuando el alumno coloca la nueva información en la parte de la memoria que corresponde a largo plazo. Las principales estrategias de adquisición de conocimientos son las lecturas y las clases. El tercero es un enfoque constructivista, basado en que el alumno construye su propio conocimiento. El modelo de Dick y Carey cuenta con 10 etapas o fases que se describen a continuación y se muestran en la figura 2.2.

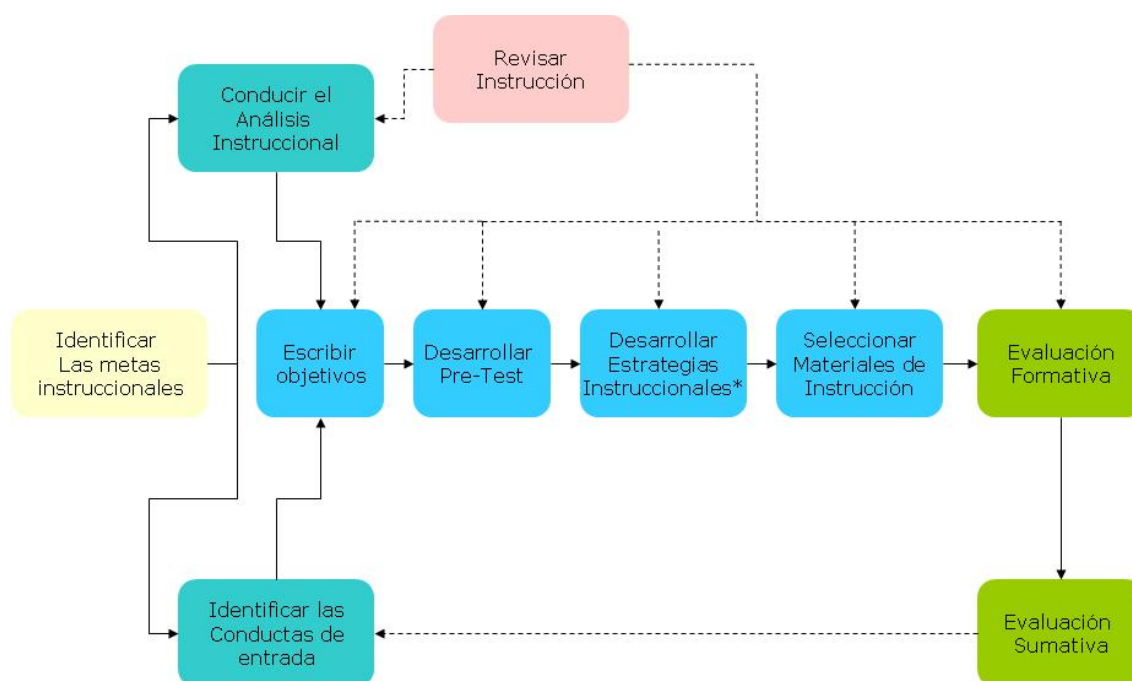


Figura 2.2: Modelo de Dick y Carey. Fuente:<http://proesa.wikispaces.com/>

1. **Identificar metas instruccionales.** Identifica qué es lo que se quiere que el estudiante sea capaz de hacer cuando se haya completado la instrucción.
2. **Llevar a cabo un análisis instruccional.** Identifica aquellas destrezas y la relación entre ellas, que deberán enseñarse para lograr la meta instruccional.

3. **Identificar las conductas de entrada.** Identifica las características de los estudiantes, el contexto en el cuál aprenderán las destrezas y el contexto en el cuál las aplicarán.
4. **Escribir objetivos.** Se describe que es lo que los estudiantes podrán hacer al terminar la instrucción, basándose en aquellas destrezas identificadas en la fase 2.
5. **Desarrollar Pre-Test.** Se elaboran las herramientas de medición para evaluar las habilidades del estudiante para lograr lo que se describió en la fase 4.
6. **Desarrollar estrategias instruccionales.** Identifica la estrategia que se utilizará para llevar a cabo la instrucción y se determine cuáles son los medios a utilizarse.
7. **Seleccionar materiales instruccionales.** Seleccionar los materiales que se utilizarán en la estrategia descrita en la fase 6. Si no se cuenta con el material adecuado para satisfacer la estrategia, se desarrolla dicho material siempre y cuando satisfagan las necesidades a los estudiantes.
8. **Evaluación formativa.** Se realizan tres tipos de evaluación (uno-a-uno, grupo pequeño y de campo), cada tipo de evaluación proporciona información para mejorar la instrucción. La información formativa puede darse en cualquier o todas las fases del modelo.
9. **Evaluación sumativa.** Se examina el valor o los méritos de la instrucción elaborada. Es una fase final donde ya se ha revisado la instrucción y conlleva a tomar una decisión sobre la efectividad del material generado, simplemente decidir si funciona o no funciona.
10. **Revisar instrucción.** Se realiza un resumen y un análisis con base en los datos recogidos de la fase 8, y se vuelven a analizar todas las fases anteriores. Finalmente, todas las revisiones se van incorporando para hacer la instrucción más efectiva.

## 2.2. Estrategias para la enseñanza

“La estrategia es un conjunto de acciones que se llevan a cabo para lograr un determinado fin”. Entonces, si llevamos esa definición a la pedagogía, las estrategias para la enseñanza son procedimientos o recursos utilizados por el docente para promover el aprendizaje significativo, el aprendizaje significativo busca que el estudiante relacione la información nueva con la que ya posee, donde va a ir reconstruyendo o modificando la información [24].

Las estrategias de enseñanza se pueden clasificar en tres categorías Preinstruccionales, Coinstruccionales y Posinstruccionales [25]:

- Las estrategias Preinstruccionales o solamente Pre, son aquellas que preparan o alertan al estudiante en relación a qué y cómo va a aprender, por ejemplo la activación de conocimiento y experiencia previa, lluvia de ideas, etc.
- Las estrategias Coinstruccionales apoyan durante el proceso mismo de la enseñanza, detectando la información principal, mantenimiento de la atención y motivación, por ejemplo ilustraciones, mapas conceptuales, analogías, entre otras.
- Las estrategias Posinstruccionales forman una visión sintética, integradora e incluso crítica del material, valorando su propio aprendizaje, como por ejemplo preguntas intercaladas, resúmenes finales, etc.

### 2.2.1. Estilos de aprendizaje

Existen diferentes tipos de estilos de aprendizaje según la teoría que se esté aplicando. Si nos basamos en una teoría de aprendizaje de acuerdo con la experiencia o la experimentación que va realizando el alumno, se clasifican cuatro estilos de aprendizaje, con base en las etapas del ciclo de aprendizaje que contempla los comportamientos de vivir la experiencia, reflexión, elaboración de hipótesis y la aplicación del conocimiento [26, 27].

- Estilo activo (Vivir la experiencia): Basada en la experiencia del alumno, son alumnos de mente abiertas, sin miedo a nuevas experiencias, con mucho entusiasmo.
- Estilo reflexivo (Reflexión): Son alumnos con facilidad de análisis, se basa principalmente en la observación y recolección de datos.
- Estilo teórico (Elaboración de hipótesis): Son alumnos metódicos, críticos, lógicos, aprenden realizando conceptualizaciones y elaborando conclusiones.
- Estilo pragmático (Aplicación del conocimiento): Se basa en desarrollar aplicaciones prácticas de las nuevas ideas. Son alumnos que experimentan, generalmente son técnicos, prácticos y realistas.

### 2.2.2. Teoría construccionista

La teoría construccionista creada por Seymour Papert se basa en la ideología “El conocimiento es construido por el que aprende”. Parte de una concepción de aprendizaje donde el conocimiento es el resultado de las experiencias personales del individuo en conjunto del trabajo propio [2]. Dando énfasis que las personas pueden aprender de formas diferentes, por consecuencia no hay estilos de aprendizaje mejores o peores simplemente son estilos.

Papert expresa que para un favorable desarrollo cognitivo, el ambiente debe estar adecuadamente organizado, estructurado y previsible. Un medio adecuado al desarrollo del alumno no debe ofrecer solamente estímulos, sino respuestas a sus acciones.

### 2.2.3. Enseñanza de la programación

En la actualidad, la educación superior es accesible para la mayoría de los jóvenes del Oeste de Europa y en Norteamérica [18]. Se ha ido incrementando la matrícula de los estudiantes en las universidades con el paso de los años. Sin embargo, en carreras referentes a computación ha tenido un efecto inverso, reduciendo el ingreso de nuevos estudiantes año con año. Asociaciones internacionales como Association for Computing Machinery (ACM), grupos especializados en la enseñanza de las ciencias computacionales como Special Interest Group on Computer Science Education (SIGCSE) y universidades de diferentes países, están investigando el por qué los jóvenes de la actualidad no tienen el interés de estudiar una carrera de computación, y por qué a los estudiantes que ingresan a estas carreras se les hace difícil aprender a programar [18]. Por estos motivos se han diseñado nuevos programas de estudio, así como, implementando nuevas estrategias y herramientas de aprendizaje.

Se cree que existen varias razones del porqué los jóvenes no les interesa estudiar una carrera de computación [15]. Como por ejemplo, que los programas de estudio no son del interés de los jóvenes, o el estereotipo que ha creado la sociedad o las películas de los programadores no motiva a los jóvenes. Por lo que, se buscan diferentes maneras para motivar a los jóvenes a estudiar una carrera referente a la computación; como por ejemplo, programando videojuegos [28] o robots educativos [29].

Por otra parte, se han aplicado cuestionarios [15, 19, 20] a diversos estudiantes de carreras referentes a la computación de diferentes universidades y maestros, preguntándoles que temas o conceptos dentro de la programación se les hace más difícil de aprender, y con esto poder modificar la estrategia de aprendizaje, el material instruccional o el programa de estudio. También, a los alumnos se les pregunta si el modo de impartir la clase por parte del profesor les ayuda o les perjudica.

Con base en lo anterior se han estado creando plataformas o herramientas para la enseñanza y motivación de programación, como por ejemplo en [15] se desarrolló una plataforma web interactiva en la programación orientada a objetos, donde se le explica al alumno por medio de una animación que estados van teniendo los objetos.

En [18] decidieron quitar las horas clase o sesión de lecturas, sustituyéndolas por horas en el laboratorio. Observaron que los estudiantes se motivan más en los laboratorios realizando prácticas, ya que piensan que la mejor manera de aprender programación es realizando el mayor número de prácticas y no de lecturas, Alyce Brady comenta que “Estudios psicológicos prueban que la mayoría de la gente aprenden más fácil,

observando ejemplos concretos para después generalizar” [30].

## 2.3. Tecnologías de la información como apoyo en la educación

La Secretaría de Educación Pública (SEP), en el Programa Sectorial de Educación, definió como uno de sus objetivos el “Impulsar el desarrollo y utilización de tecnologías de la información y la comunicación en el sistema educativo, para apoyar el aprendizaje de los estudiantes, ampliar sus competencias para la vida y favorecer su inserción en la sociedad del conocimiento”. Además, define como una de sus acciones el promover la utilización de espacios virtuales para acercar a docentes y estudiantes a la tecnología de punta, así como desarrollar competencias para su uso [31].

Actualmente, con el uso de nuevas tecnologías, tales como Internet, realidad virtual y sistemas inteligentes, es posible tener ambientes de aprendizaje con una mayor impacto educativo [8]. Este beneficio, puede llegar a estudiantes de cualquier nivel educativo y socio-económico; apoyando en la reducción de la brecha digital respecto al uso de estas tecnologías y al proceso de enseñanza-aprendizaje.

El principal problema en escuelas que desean utilizar esta tecnología es contar con la capacidad económica y la infraestructura adecuada. Además, estudios que se han realizado en instituciones donde se utiliza este tipo tecnología como herramienta de apoyo en la formación de sus alumnos, muestran que el aumento del aprendizaje del estudiante depende principalmente de la capacidad y el interés del maestro al utilizar estas tecnologías [32].

Por otra parte, un punto que favorece la adquisición de competencias digitales, es que nuevas generaciones de estudiantes están acostumbrados en aprender, crear y comunicarse usando tecnología. Estudios recientes indican el gran número de personas que utilizan tecnología en su vida cotidiana; principalmente niños y jóvenes estudiantes [32]. Con base en lo anterior, se busca aprovechar el conocimiento o el acercamiento que ya se tiene a la tecnología, y así darle un contexto educativo en el cual los estudiantes se enfocan en aprender, incrementando la comprensión y la abstracción de conceptos.

### 2.3.1. Robótica pedagógica

La robótica educativa es un área que tiene amplias posibilidades de posicionarse como herramienta de construcción de conocimiento. La robótica educativa, en conjunto con el uso de aprendizaje colaborativo, abordaje por proyecto y construcción colectiva, aportan el marco metodológico de referencia que concretiza tales potencialidades [22]. Existe una diversidad de investigaciones que estudian el impacto de esta área de conocimiento. Además que, al utilizar robots pedagógicos favorece la utilización del método

intuitivo, considerado uno de los mejores métodos para enseñar [33]. A continuación, se describen algunas trabajos relevantes sobre el tema.

En [20] se utilizó LEGO Mindstorm como herramienta de aprendizaje. Tomando como referencia la facilidad de armar robots y manipular los movimientos de estos, los estudiantes percibían estar en un ambiente de juego, en donde desarrollaban su imaginación, creatividad, razonamiento lógico, fortalecimiento de habilidades para resolver problemas, así como, el trabajo en equipo [2].

En [34] se ha trabajado con LEGOS para la enseñanza de lógica programática. A la par con las actividades de difusión y divulgación de estos programas, se ha desarrollado el software denominado KareLEGO, un prototipo inspirado en el lenguaje de programación del Robot Karel que manipula físicamente un Robot Mindstorm NXT. La finalidad de KareLEGO es darle un sentido práctico a la asignatura de Diseño de Algoritmos que actualmente se imparte de manera obligatoria en el Tronco Común de las licenciaturas que oferta la Facultad de Ciencias de la Universidad Autónoma de Baja California. Tanto en esta asignatura, como en otras, se ha observado que a los alumnos se les dificulta la programación, debido a todos los conceptos abstractos inherentes a los lenguajes de programación. Con KareLEGO, los estudiantes encuentran una aplicación en la vida real, viendo respuestas físicas y tangibles que puede desarrollar el robot Mindstorm [34]. Sin embargo, para poder utilizar este programa se debe estar en el laboratorio físicamente.

En [35] se desarrolló un prototipo de robot educativo, con el objetivo de fortalecer la rama de la robótica pedagógica en Latinoamérica. Este prototipo se trata de un robot móvil autónomo, el cual integra un microcontrolador, motores y sensores. Para la programación de este robot se utiliza una interfaz gráfica sencilla e intuitiva basada en lenguaje Pascal. Esta interfaz posee todas las herramientas necesarias para la creación, edición y visualización de los programas mediante diagramas de bloques. Junto con el prototipo y el ambiente gráfico de programación, los estudiantes aprenden principios y teorías de la programación.

En [36], también utilizaron robots LEGO Mindstorm como herramientas que motivaran en el aprendizaje de programación en lenguaje C/C++; dirigido a estudiantes de educación superior. Esto debido a que los estudiantes se quejaban continuamente diciendo que programar era aburrido o sentían que no eran buenos. Los estudiantes utilizaron diferentes librerías de este lenguaje para programar al robot, destacando las librerías RobotC para principiantes y NXT++ para estudiantes que ya tenían una noción de programación.

### 2.3.1.1. Juego de robótica educativa

Existen diferentes kits educativos y comerciales, como por ejemplo, Fischertechnik, Kit de robótica de Arduino, Kit de robótica de Parallax, Robotis Bioloid, Nao, LEGO Mindstorm, ver figura 2.3. Cada uno de estos kits presentan diferentes características, algunos son humanoides, carros y con otros se puede armar diferentes modelos por sus piezas intercambiables. Por lo general estos robots tienen un microcontrolador y memoria donde se guarda el programa para realizar las acciones del robot, unos con más capacidad que otros. Algunos robots tienen la característica de poder conectar varios sensores y servomotores los cuales son programables, pero no todos pueden comunicarse inalámbricamente con otros dispositivos (Bluetooth, WI-FI). Existen diversos lenguajes con los que se pueden programar cualquiera de los robots, pero se destaca el robot de LEGO por tener la flexibilidad de programarse con distintos lenguajes como por ejemplo: C, C++, Java, C#, entre otros [37, 38].

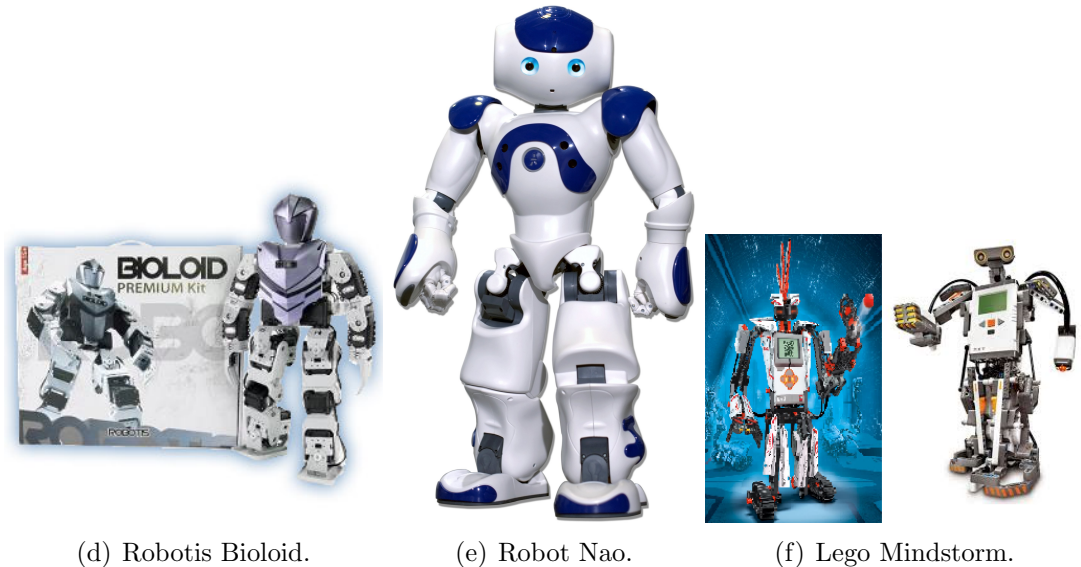


Figura 2.3: Ejemplos de robots pedagógicos.

## 2.4. Laboratorios virtuales y teleoperados

El contar con laboratorios de prácticas se encuentra tradicionalmente limitado por factores económicos, espaciales y temporales. Económicos por el elevado costo de equipos de vanguardia y la disponibilidad del recurso humano capacitado. Espaciales porque no se cuenta con suficientes salas de laboratorio para atender la totalidad de la población estudiantil, y la realización de estas prácticas se limita al horario establecido para clase. Finalmente, la limitante temporal es debido a que los laboratorios son de acceso restringido, a no ser que se encuentre disponible un profesor o supervisor a cargo [39].

Debido a estas limitaciones, es que se ha tenido un avance considerable en la investigación del uso de las tecnologías de la información y de la comunicación para derribar la barrera existente o, por lo menos, reducirla en la medida de lo posible [39]. Una alternativa utilizada es el uso de laboratorios virtuales y/o teleoperados. Los laboratorios virtuales son sistemas computacionales accesibles vía Internet, donde se puede simular el funcionamiento de un laboratorio tradicional.

En algunas instituciones educativas se han implementado este tipo de laboratorios atendiendo diferentes áreas de conocimiento. Algunos ejemplos son: 1) RECOLAB, laboratorio remoto de control basado en MATLAB y SIMULINK, este laboratorio proporciona una arquitectura general para la simulación de esquemas de control y su ejecución remota en tiempo real sobre maquetas físicas [10]; 2) Laboratorio virtual de química, donde se simula todo el espacio físico del laboratorio, como las puertas, gabinetes, mesas de trabajo, ventanas, bancos, etc., buscando que el estudiante a distancia tenga una experiencia lo más real posible, como si estuviera dentro del entorno real. Esta interacción se lleva a cabo antes de tener su primera sesión en el laboratorio físico [40]; 3) Laboratorio virtual para el área de Electrónica, Automatización y Control, es una herramienta para iniciarse en el análisis y diseño de circuitos, facilita la generación de ambientes de aprendizaje basados en experimentos de laboratorio con problemáticas reales [41]; 4) Laboratorio remoto para el curso de comunicaciones análogas y digitales con acceso vía Internet. El estudiante cuenta con un interfaz gráfica basada en LabView, donde puede manipular equipo real para el análisis de señales [42].

### 2.4.1. Laboratorios virtuales y teleoperados de robótica

En el estado del arte de los laboratorios virtuales y teleoperados de robótica, se han encontrado como casos de éxito los siguientes:

En [8] se desarrolló ROBOLAB, un ambiente educativo en línea que consta de un laboratorio virtual y teleoperado de un brazo robótico. Este sistema permite manipular remotamente el dispositivo con flexibilidad de horarios. Este proyecto fue realizado para la enseñanza de conceptos de robótica, cinemática y programación, dirigido a alumnos de Ingeniería en Informática de la Universidad de Alicante. El desarrollo de este proyec-

to se originó por la demanda de un gran número de estudiantes para utilizar el brazo. ROBOLAB cuenta con dos tipos de retroalimentación para el alumno. La primera es una simulación en 3D del movimiento del brazo. En la segunda forma, el alumno recibe retroalimentación a través de vídeo en vivo. Se logró un impacto favorable en la formación de los estudiantes con un 81 %. De igual manera, alumnos que utilizaron ROBOLAB indicaron que les agradó el poder realizar sus prácticas desde casa y con una flexibilidad de horarios.

En [11] se presenta un laboratorio remoto que ha sido desarrollado con el objetivo de que los alumnos puedan hacer uso a través de Internet de los robots móviles disponibles en el laboratorio. Dispone de un equipo heterogéneo de diferentes robots móviles que el alumno debe poder utilizar para poner a prueba los algoritmos desarrollados. Donde también se debe permitir la interacción de un robot con el resto de los mismos llevando a cabo una monitorización de todos ellos mientras se está ejecutando una tarea. Cada robot debe poder interactuar con los algoritmos que están corriendo en el resto de los robots activos en la plataforma. La realización de prácticas a distancia permite que los alumnos puedan realizar experimentos en cualquier momento del día y cualquier día que los dispositivos físicos se encuentren operando, incluidos los fines de semana sin atender a un horario previamente planificado.

El laboratorio en [9] tiene como objetivos el minimizar costos en la formación del estudiante y que el estudiante en línea tenga una experiencia educativa equivalente a un estudiante presencial. Este laboratorio virtual de robótica tiene que ser accesible de forma remota, y utilizar realidad virtual para ver una simulación en respuesta de lo que el estudiante programó.

En [12] se desarrolló un laboratorio teleoperado de robótica, con el objetivo de resolver el problema de educación a distancia y manipular robots reales desde su casa con conexión a Internet y visión en tiempo real, para alumnos de Maestría de Ingeniería en Sistemas y Control Automático. Para poder utilizar este laboratorio se enfrentaron con 3 dificultades, la interfaz del laboratorio debe de ser intuitiva y el lenguaje que se va utilizar debe de ser conocido por los estudiantes, la segunda dificultad es el poco equipo que hay de robots por su costo elevado y la tercera es tener un horario establecido porque en los laboratorios hay más estudiantes que equipo. Para resolver la primera dificultad se utilizó MATLAB y EJS como las herramientas para la interfaz del laboratorio. Para superar la segunda dificultad, se decidió utilizar los robots de LEGO y por último, la tercera dificultad, se resolvió a través de una agenda electrónica donde el estudiante puede revisar que laboratorio esta desocupado.

A diferencia de los laboratorios [8, 12, 11, 9], el trabajo que se propuso se centró en la formulación de prácticas bien estructuradas de programación, con un enfoque instruccional, con la finalidad de apoyar en cursos de la enseñanza de lógica programática. Además se creó la comunicación entre el servidor web y el servidor del robot, así mis-

mo se diseño y desarrolló el ambiente de programación del alumno para completar los diagramas de flujo y el ambiente para generar prácticas por parte del maestro.

## Capítulo 3

# Diseño de prácticas instruccionales

Buscando dar respuesta a las preguntas de investigación que atiende nuestra propuesta, en este estudio se utilizaron métodos y procedimientos basados en los enfoques de investigación cuantitativa y cualitativa [43]. La utilización de un enfoque mixto de investigación es derivado de nuestro interés en medir cuantitativamente el impacto en el desempeño educativo de los sujetos de estudio, complementado con el análisis de experiencias y emociones experimentadas tanto por estudiantes como de los instructores participantes en el estudio. Los datos obtenidos mediante la combinación de ambos enfoques permitió obtener una apreciación más clara y precisa sobre el impacto real de los materiales instruccionales generados.

Para el enfoque de investigación mixto, se tomaron las siguientes consideraciones: (1) ambos métodos de investigación, cualitativo y cuantitativo, tienen igual nivel de prioridad; (2) la secuencia de implementación de los métodos fue en forma paralela, concurrentemente se colectaron y analizaron los datos; y (3), la integración de los datos cualitativos y cuantitativos, también se realizó en el momento en que estos fueron recolectados. A continuación se describe como se realizó la selección de temáticas para el LTRP y la guía de generación de material instruccional, además se describen todas las versiones que se diseñaron del ambiente de programación.

### 3.1. Selección de temáticas

Aún cuando asumimos que en el LTRP se podrían implementar una diversidad de ejercicios, atendiendo las temáticas de programación con mayor problema de aprendizaje por los estudiantes, se consideró que al utilizar el ambiente propuesto, algunas de estas temáticas son más viables de impactar significativamente en su aprendizaje. Por consiguiente, fue de nuestro interés identificar las temáticas que, al enseñarse a través del LTRP, incrementasen el aprovechamiento educativo de los estudiantes señalados en [19, 20].

### 3.1.1. Identificación de temáticas objetivo

Para determinar los temas o conceptos de programación con mayor problemática para su enseñanza, se consideraron aquellos ya identificados e indicados en la literatura [19, 20]. Adicionalmente, se ratificó esta lista de temas con la opinión de alumnos y maestros de nuestra institución. El punto de vista de nuestra audiencia objetivo, se obtuvo a través de una entrevista con los profesores y la aplicación de una encuesta dirigida a los alumnos. En el apéndice A, se presenta la encuesta que se aplicó para este fin.

### 3.1.2. Materiales

Los instrumentos que se utilizaron en el levantamiento de datos están basados en los propuestos por [19, 20], para la identificación de temáticas complejas en el área de programación. La adaptación de este instrumento al ambiente de los sujetos de estudio, se presenta a detalle en el apéndice A. Este cuestionario permite corroborar cuales son las temáticas que más se les dificulta para poder implementarse en el LTRP.

### 3.1.3. Selección de participantes

Se seleccionaron dos profesores que habían impartido cursos de introducción a la programación o diseño de algoritmos en por lo menos tres ocasiones. Y se seleccionaron 36 alumnos que estuvieron cursando en alguna de las materias mencionadas anteriormente.

### 3.1.4. Temáticas factibles de implementación

Atendiendo la primer pregunta de investigación, ¿Cuáles son las temáticas de programación que pueden ser implementadas a través de prácticas interactivas en el LTRP?, se eligieron los temas que presentan más dificultad de acuerdo a las deficiencias de los alumnos de primeros semestres de la carrera de Ciencias Computacionales. A continuación se muestra algunos de los resultados obtenidos de la encuesta que se les aplicó a los alumnos.

- ¿Qué se les dificulta a los alumnos?

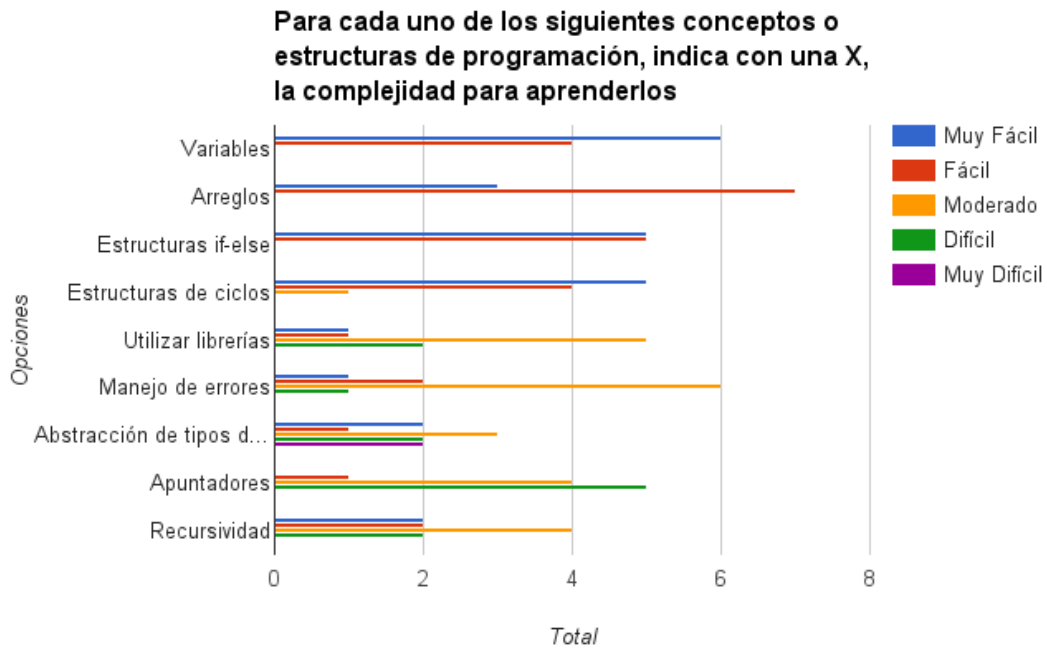


Figura 3.1: Respuestas de la pregunta. ¿Qué se les dificulta a los alumnos?

- ¿Cuándo crees que aprendes más? Sesiones de prácticas, estudiar por tu cuenta.

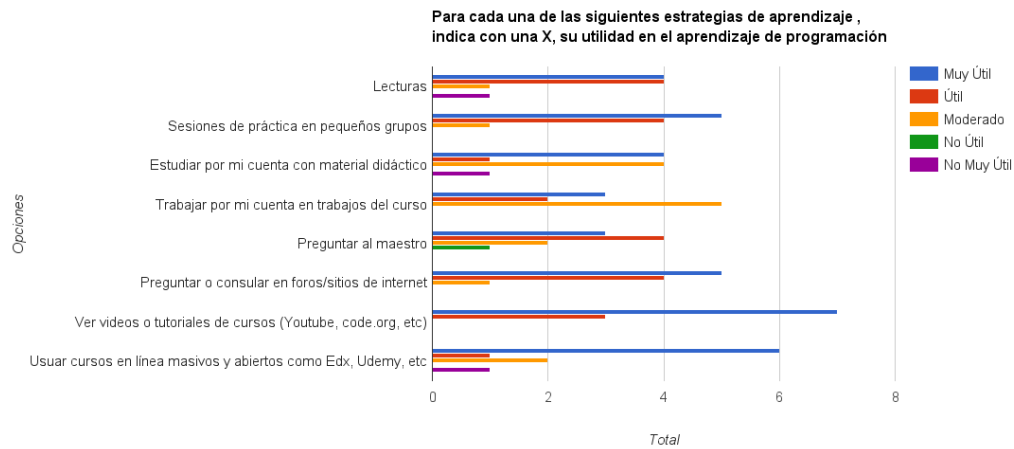


Figura 3.2: Respuestas de la pregunta. ¿Cuándo crees que aprendes más? Sesiones de prácticas, estudiar por tu cuenta.

- ¿Qué material crees que te ayuda para aprender a programar? Material interactivo, programas de ejemplo, ejercicios de preguntas y respuestas.

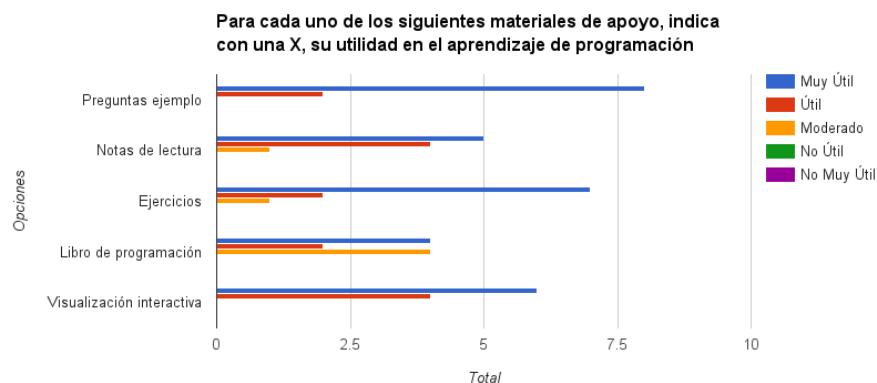


Figura 3.3: Respuestas de la pregunta. ¿Qué material crees que te ayuda para aprender a programar? Material interactivo, programas de ejemplo, ejercicios de preguntas y respuestas.

Con respecto a las respuestas presentadas en las figuras anteriores 3.1 a la 3.3, los alumnos tienen dificultades con temas como apuntadores, abstracción de datos, recursividad, estructuras de control, por lo que se decidió seleccionar el tema de estructuras de control y adaptarlo en cada uno de las actividades que se desarrollaron en la tesis. Además, en la figura 3.2 se puede observar que una de las estrategias de aprendizaje con más porcentaje son las que el alumno puede desarrollar por su cuenta. Respecto a los materiales de apoyo presentados en la figura 3.3, se pudo observar que un gran porcentaje son las preguntas o ejercicios de ejemplo, resolver ejercicios y actividades interactivas. Gracias a las respuestas obtenidas se determinó la factibilidad técnica (ej. movimientos del robot, uso de sensores, área de desplazamiento, entre otros) para el desarrollo de las prácticas que se agregaron al LTRP. Lo anterior se determinó por el proponente de esta investigación y sus asesores. Otro resultado de esta fase, es el diseño de una guía técnica para la elaboración de material instruccional utilizado en laboratorios teleoperados de robots pedagógicos; potencialmente transferible a ambientes presenciales de instrucción mediante este tipo de robots.

### 3.1.5. Implementación de ejercicios

En esta fase se definieron una serie de ejercicios a realizar y las competencias que se deseaban cubrir. Siguiendo la metodología de diseño instruccional propuesta por Dick y Carey [21], se diseñó e implementó un guía para los maestros, donde se explica paso por paso como generar material instruccional para el LTRP (Ver el apéndice B: Guía para el diseño de material instruccional basado en el modelo de Dick y Carey). El modelo de Dick y Carey establece una metodología para el diseño de la instrucción con base en un modelo reduccionista de la instrucción al seccionarla en pequeños componentes (diez componentes o etapas, que se explican en el capítulo 2). La instrucción se dirige específicamente en las habilidades y conocimientos que se enseñan y proporciona las

condiciones para el aprendizaje.

La guía diseñada se utilizó para generar el contenido educativo que se implementó en todas las actividades que se desarrollaron durante el trabajo de tesis. A continuación se explica que se realizó en cada una de las diez etapas que componen este modelo.

Como primera etapa se identificaron las metas instruccionales. Para poder identificarlas, inicialmente se planteó el problema a tratar en este estudio, el cual fue que a los estudiantes de cursos introductorios de programación se les dificulta aprender algunos conceptos básicos, entre los que se encuentran la construcción y utilización de estructuras de control. Teniendo como propósito definir el material instruccional que facilite al estudiante comprender estos temas, se propusieron las siguientes metas.

- Familiarizarse con conceptos de estructuras de control.
- Buscar que los estudiantes adquieran habilidades para construir y aplicar efectiva y eficientemente estructuras de control.

En la segunda etapa, se describen las habilidades y destrezas que los alumnos van a desarrollar y las que van a fortalecer en forma de actividades. Continuando con nuestro diseño, con base en las metas descritas en la etapa uno, se definió que las destrezas o habilidades de los alumnos es aprender a construir estructuras de control y desarrollar el razonamiento para saber donde utilizarlas, a su vez, fortaleciendo el conocimiento ya adquirido de diseñar algoritmos.

Se plantearon los objetivos de nuestra instrucción y como cumplirlos, por ejemplo, para la actividad “Estructuras de control: Selectivas” el objetivo es que el alumno aprenda a utilizar los operadores relacionales y lógicos para construir correctamente la condición dentro de una estructura selectiva (if-then, if-then-else). Para cumplir este objetivo se realizó una clase teórica sobre como utilizar los operadores para construir correctamente la condición dentro de una estructura selectiva y después se realizaron las prácticas.

Lo siguiente fue realizar una estrategia instruccional para alcanzar cada uno de los objetivos propuestos. Lo primero que se hace es describir los medios o instrumentos a utilizarse para completar el objetivo y lo segundo es determinar la cantidad de tiempo que llevaría a cabo en realizar cada actividad. Para esta etapa en nuestro estudio se diseñó y desarrolló un pre-test y un post-test, los cuales se pueden revisar en el apéndice C. El pre-test fue un examen de diagnóstico para identificar el conocimiento sobre diseño de algoritmos y lógica programática que tienen los alumnos, y como post-test se realizó una sesión de prácticas sobre estructuras de control selectivas y posteriormente se aplicó el mismo examen de diagnóstico que se utilizó en el pre-test, lo anterior es recomendado para comparar los resultados entre los dos exámenes. El tiempo estimado de solución de las prácticas por un alumno fue de 20 a 30 min.

Por último tenemos la etapa de evaluación formativa, en esta parte se analizaron los comentarios por parte de los alumnos y los maestros en cada una de las fases que se desarrollaron en este estudio para ir mejorando la instrucción.

Utilizando la guía instruccional se diseñaron las prácticas a utilizarse en actividades diseñadas para fortalecer el conocimiento del alumno en tu tema concreto, en este caso estructuras de control, empleando diferentes materiales y herramientas. A continuación se describen las actividades diseñadas entre las que se destaca “Simulando un robot”, “Programando en papel” y “Ambiente web y programación con SCRATCH”.

## 3.2. Actividad 1: “Simulando un robot”

Esta actividad, diseñada por el alumno Ramón Ramirez de la Facultad de Ciencias, está basada en Karel el robot. Karel el robot es una aplicación sencilla que consta de un simulador de robot integrado para probar instrucciones de programación. Este programa es comúnmente utilizado en alumnos que están aprendiendo a programar, ya que provee bases sobre lo que significa diseñar un programa de instrucciones aplicable a diferentes situaciones.

La actividad se realiza en equipos donde un integrante actúa como el programador del robot y el otro actúa como el robot. El programador desarrolla el algoritmo mientras que el integrante que actúa como robot tiene los ojos tapados, para que solamente siga las instrucciones que el programador le va diciendo (figuras 3.4, 3.5). La práctica consiste en que el robot tiene que llegar a recoger un zumbador que se localiza en algún lugar del ambiente que lo rodea, al programador se le da un tiempo para que diseñe el algoritmo que cumpla con el objetivo de recoger el zumbador, lo interesante de esta práctica es que el programador solo puede utilizar un número limitado de palabras reservadas (figura 3.6) y colocarlas en el pizarrón de prácticas (figura 3.7, 3.8), donde las instrucciones deben colocarse en orden. Una vez terminado el algoritmo, se le dicta al robot cada una de las instrucciones que tiene que realizar y verificar si el algoritmo cumple con el objetivo. Por ejemplo en la figura 3.9, se muestra mediante un diagrama la ruta que debe de realizar el robot donde debe avanzar 8 pasos, girar a la derecha, avanzar 5 pasos, agacharse (que incluye recoger zumbador), levantarse y regresar a su posición original. En la figura 3.8 se muestra el código del ejercicio.

Esta actividad se presentó los alumnos de primer semestre de la Facultad de Ciencias como parte del material de apoyo en la materia de Diseño de Algoritmos, participaron 26 alumnos y realizaron diferentes ejercicios con diferentes grados de dificultad.



Figura 3.4: Ejecutando el algoritmo I.

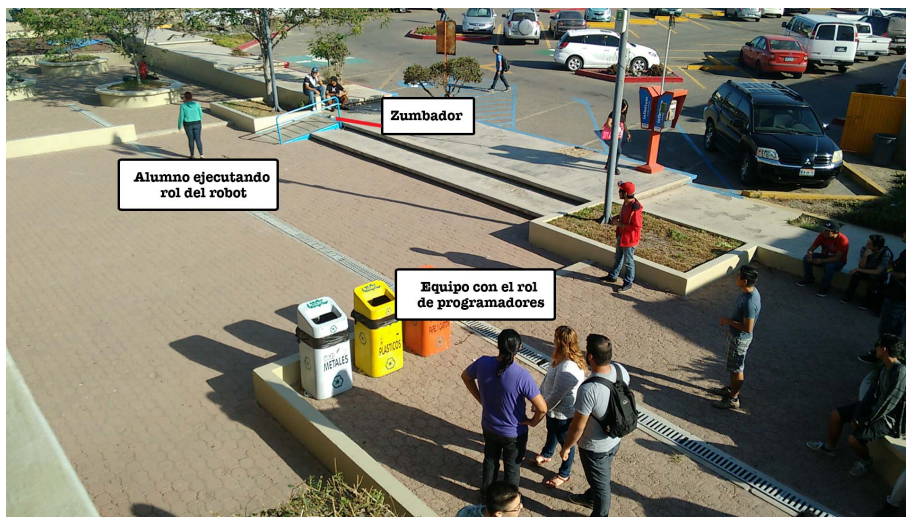


Figura 3.5: Ejecutando el algoritmo II.

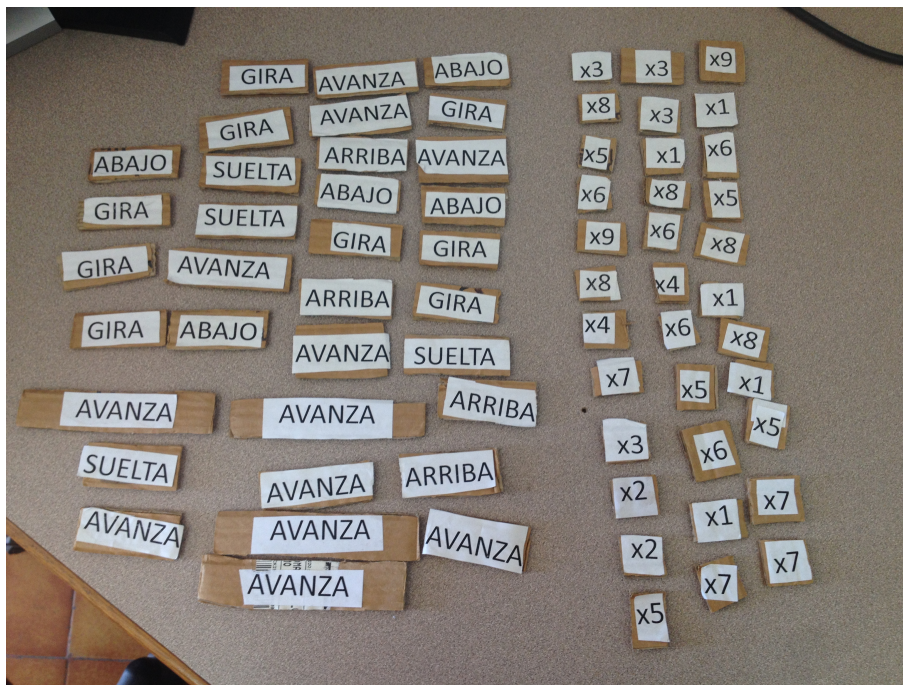


Figura 3.6: Palabras a utilizar.



Figura 3.7: Pizarrón de prácticas.

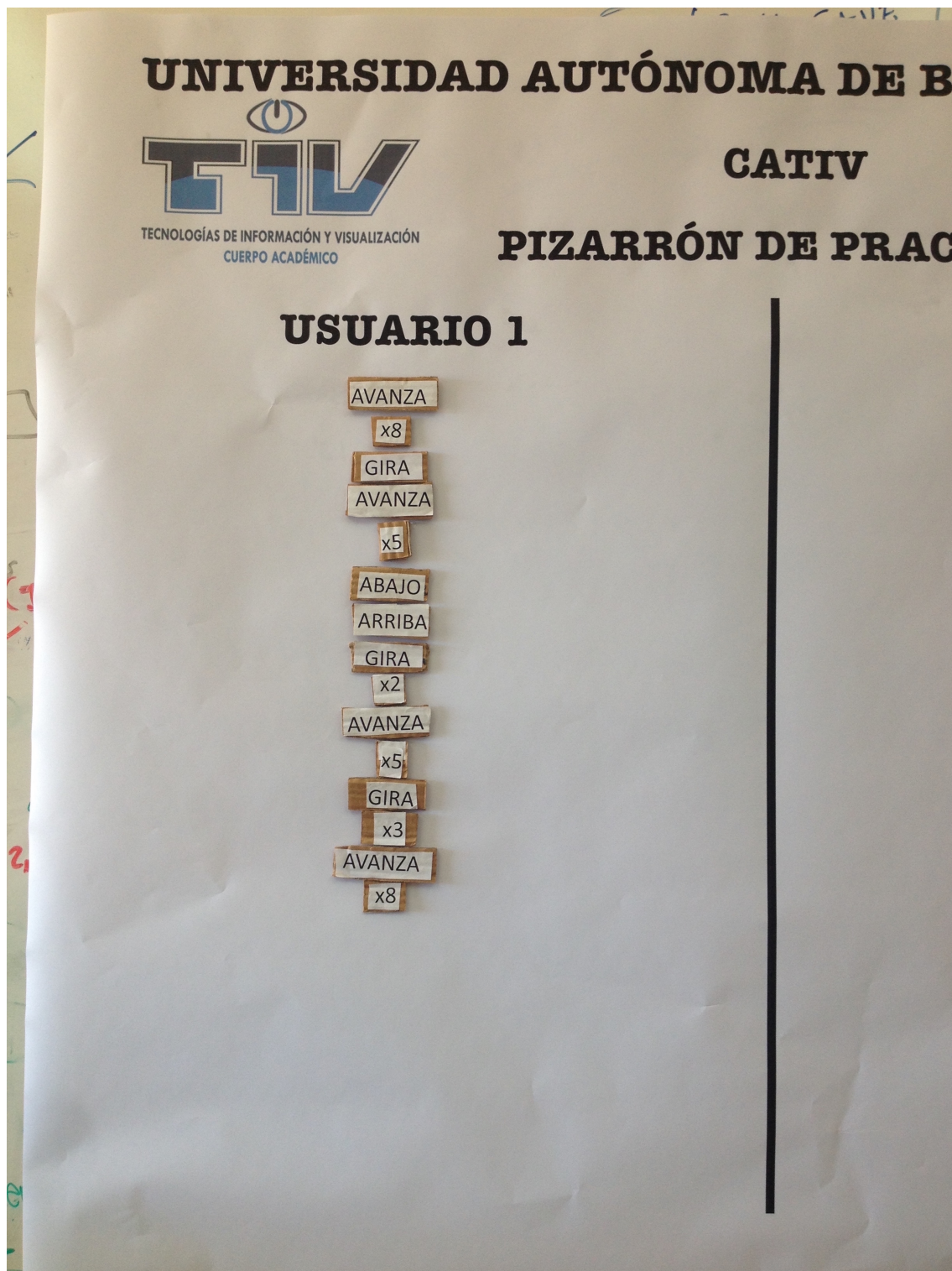


Figura 3.8: Ejemplo de práctica.

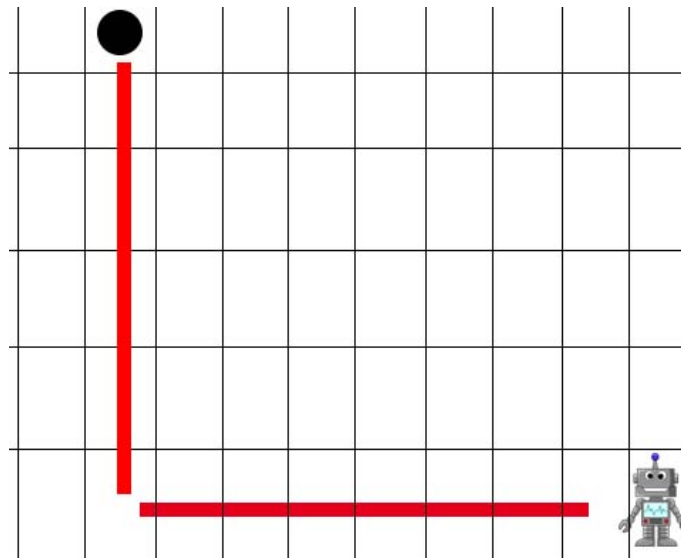


Figura 3.9: Diagrama ruta del robot.

### 3.2.1. Objetivo

Con la temática previamente seleccionada, esta actividad busca que el estudiante adquiera habilidades y destrezas básicas de lógica programática al desarrollar programas basados en el mundo de Karel. Enfatizando que esta actividad se realiza sin la utilización de una computadora.

### 3.2.2. Resultados

Esta actividad se centró en obtener resultados de aceptación y motivación por parte de los estudiantes, lo anterior se midió mediante la aplicación de una encuesta cuyos resultados se resumen en la tabla 3.1.

Tabla 3.1: Encuesta aplicada a los participantes de la actividad Simulando un robot

Pregunta	Me gustó	Indiferente	No me gustó		
Pregunta	Muy de acuerdo	De acuerdo	Normal	Desacuerdo	Muy en desacuerdo
¿Qué te pareció la actividad?	84.6 %	15.4 %	0 %		
¿Te ayudó a comprender el tema visto en clase?	53.8 %	27 %	15.4 %	3.8 %	0 %
¿La actividad te motivó a corregir errores cuando no cumplías el objetivo?	53.8 %	27 %	19.2 %	0 %	0 %
¿Crees que este tipo de actividades te ayude en tu lógica programática?	42.3 %	27 %	23 %	7.7 %	0 %
¿Crees que es necesario utilizar una computadora para aprender lógica programática?	23 %	19.2 %	11.6 %	27 %	19.2 %

Como podemos observar en la tabla 3.1, esta actividad tuvo buena aceptación por parte de la mayoría de los participantes, ya que es dinámica y divertida, rompiendo un poco con la monotonía de estar sentado en el mesabanco o enfrente de la computadora, esto ayudó a que los alumnos se motivaran al querer cumplir con los objetivos de las prácticas. Algunos de los participantes nos compartieron los siguientes comentarios “La actividad fue muy emotiva y divertida, no estaría mal usarla con niños para introducirlos a este tipo de ciencia”, y “La actividad me gustó bastante hubiera querido que la aplicáramos en clase para hacer los ejercicios más dinámicos y ayudarnos a una mejor comprensión y elaboración del programa Karel”. Sin embargo, al preguntarles si la computadora es una necesidad para aprender lógica programática, hubo opiniones encontradas donde el 42.2 % de los participantes piensa que la computadora es necesaria, el 46.2 % cree que para este nivel no es una necesidad y si pueden utilizar actividades como el “Robot Humano” para ir adquiriendo conceptos básicos de lógica programática.

También se le aplicó una encuesta al maestro encargado de impartir la materia de Diseño de Algoritmos, esto debido a que se deseaba obtener retroalimentación de la actividad para saber si el maestro la utilizaría como herramienta de apoyo dentro de su planeación, así como obtener comentarios constructivos para modificar la actividad. A continuación se muestra la encuesta con las respuestas del maestro.

1. ¿Qué le pareció la actividad? *“En términos generales bien, se cumplió con el objetivo, el cual era que los estudiantes ejecutaran las instrucciones del robot”.*
2. ¿Cree que este tipos de actividades didácticas le ayude a la enseñanza de algoritmos?¿Por qué? *“Si, las instrucciones son más claras para el alumno, además se fortalecen los conceptos de orden, secuencia, instrucción y lógica”.*
3. ¿Percibió motivación e interés por parte de los alumnos al realizar la actividad? *“Si, hay una motivación extra cuando se trata de colaborar para realizar una tarea, se conjuntan ideas y se construyen las soluciones más fácilmente”.*
4. De la manera más concreta explique ¿Qué le gustó de la actividad y que cambiaría de la misma? *“Me gusto que tuvieron el interés de participar en la actividad. Solo agregaría un par de ejercicios para complementar la actividad incorporando nuevas instrucciones”.*
5. Con los cambios propuestos por usted ¿Incluiría este tipo de actividades en su planeación para la enseñanza de diseños de algoritmos? *“Si, aunque no serían muchas actividades solo al inicio para reafirmar algunos conceptos”.*

### 3.2.3. Conclusión

Con base en las respuestas o comentarios obtenidas por parte de los alumnos y el maestro se concluye de manera preliminar que al realizar este tipo de actividades que complementan la teoría vista en clase, rompe la monotonía del aula, obligando a los alumnos a moverse y no estar estáticos frente a una computadora. Esto implica que las prácticas sean dinámicas y lúdicas, estimulando la participación de los alumnos para desarrollar habilidades y valores como trabajo en equipo, respeto, comunicación oral, etc. Sin embargo, se recomienda realizar estudios más amplios que complementen los resultados obtenidos.

## 3.3. Actividad 2: Diagramas de flujo y programación con Scratch

La segunda práctica implementada, se basa en un ambiente en papel fácil de utilizar, donde se muestra al alumno un diagrama de flujo incompleto, la meta es completar el diagrama con algunas opciones que se le muestran, una vez que tiene el diagrama terminado tiene que seguirlo paso por paso para construir el programa en el lenguaje Scratch (figura 3.16 a la 3.18). Scratch es un lenguaje de programación diseñado y desarrollado por Mitchel Resnick, permiten que los jóvenes exploren y experimenten con los conceptos de programación mediante el uso de una sencilla interfaz gráfica. El tema central de las prácticas es que los alumnos adquieran el conocimiento de las estructuras de control, como su construcción y el uso de las mismas. En las figuras 3.10

a la 3.12 se pueden observar las prácticas que se diseñaron, y en las figuras 3.13 a la 3.15 se muestran las prácticas con sus respuestas. A continuación se describe en que consiste cada una de las prácticas.

- **Práctica 1.** Primeramente, se les da a los alumnos una descripción de ella, además se les proporciona indicaciones de cada paso para diseñar el algoritmo y así se espera puedan completar los diagramas. En esta práctica se imitará el juego de “GUACA-MOLE”. Este juego trata de ir golpeando con martillo a un topo que se asoma y se esconde, pero en esta práctica consiste en clicar al extraterrestre “Gobo” (figura 3.10) antes de que desaparezca. El programa del juego debe ir aumentando la dificultad cuando se cumpla un número de aciertos, cada acierto implica que el usuario da un click con el ratón encima de Gobo, aumentado la velocidad de desaparición de Gobo. Al menos se debe de aumentar 2 niveles de dificultad. Y las indicaciones para cumplir el objetivo son las siguientes. En la figura 3.16 se muestra el código completo en Scratch de la práctica.

1. Si hacemos click en Gobo entonces la variable “score” debe de sumarse 1, y reproducirá un sonido indicando que ha sido presionado.
2. Crear una variable que se llame “score”.
3. Al iniciar el programa la variable debe ser igual a 0.
4. Posicionar a Gobo en las coordenadas  $x=0$ ,  $y=0$ .
5. Antes de comenzar el programa debe de mostrarse Gobo y un mensaje de inicio por 2 segundos.
6. Como este juego no tiene fin, hay que seleccionar el ciclo correcto para que siempre corra el programa.
7. Ahora se debe de esconder a Gobo y que espere 1 segundo, para posicionarse en las coordenadas  $(x,y)$  al azar.
8. La coordenada “x” debe estar entre -200 y 200 y la coordenada “y” debe de estar entre -140 y 140.
9. Una vez que seleccione su nueva coordenada Gobo volverá aparecer.
10. Si queremos que a una cantidad específica del “score” Gobo se esconda más rápido se tiene que hacer los siguiente:
  - a) Debemos de preguntar si Score es mayor a 5 y menor a 11, si es verdad entonces solo debe esperar por 0.5 segundos.
  - b) Sí score es mayor a 10 entonces Gobo solamente se espera por 0.2 segundos.
  - c) En el caso de que no sea mayor a 5, ni mayor a 10 entonces se mostrará por 0.7 segundos.

**XXXI Semana de Ciencias,  
Taller de animaciones con Scratch  
del 21 al 24 de octubre**

Para realizar los ejercicios, sigue las indicaciones del facilitador. Por favor proporciona la siguiente información:

Edad: \_\_\_\_\_ Grado: \_\_\_\_\_ Escuela: \_\_\_\_\_ Fecha: \_\_\_\_\_

Conocimientos de computación: si  no

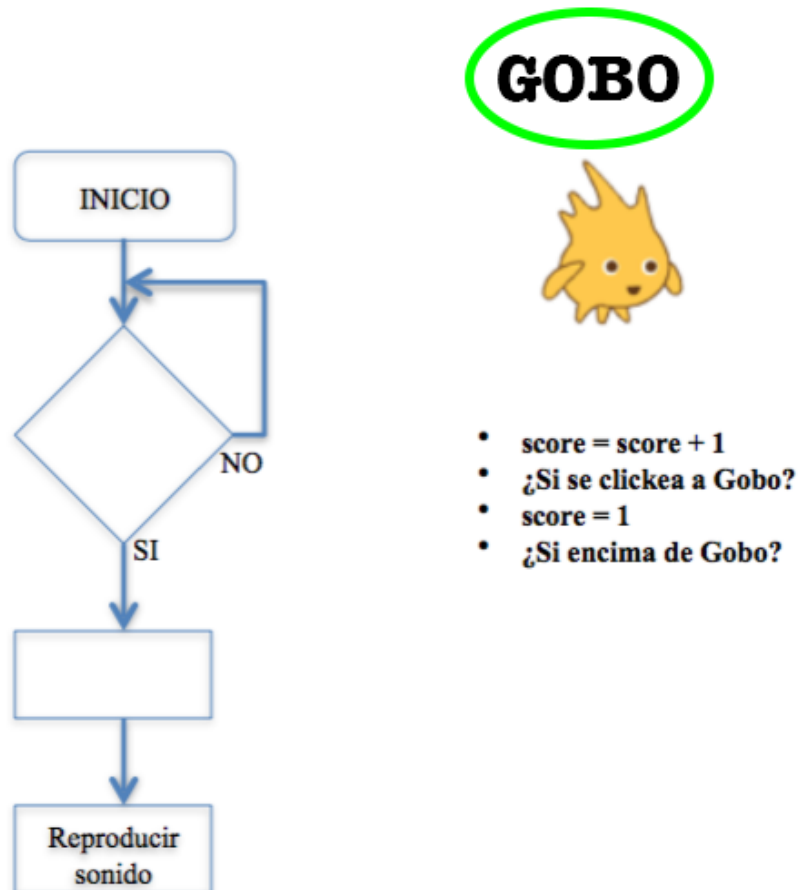


Figura 3.10: Práctica 1 primera parte. Nivel de dificultad bajo.

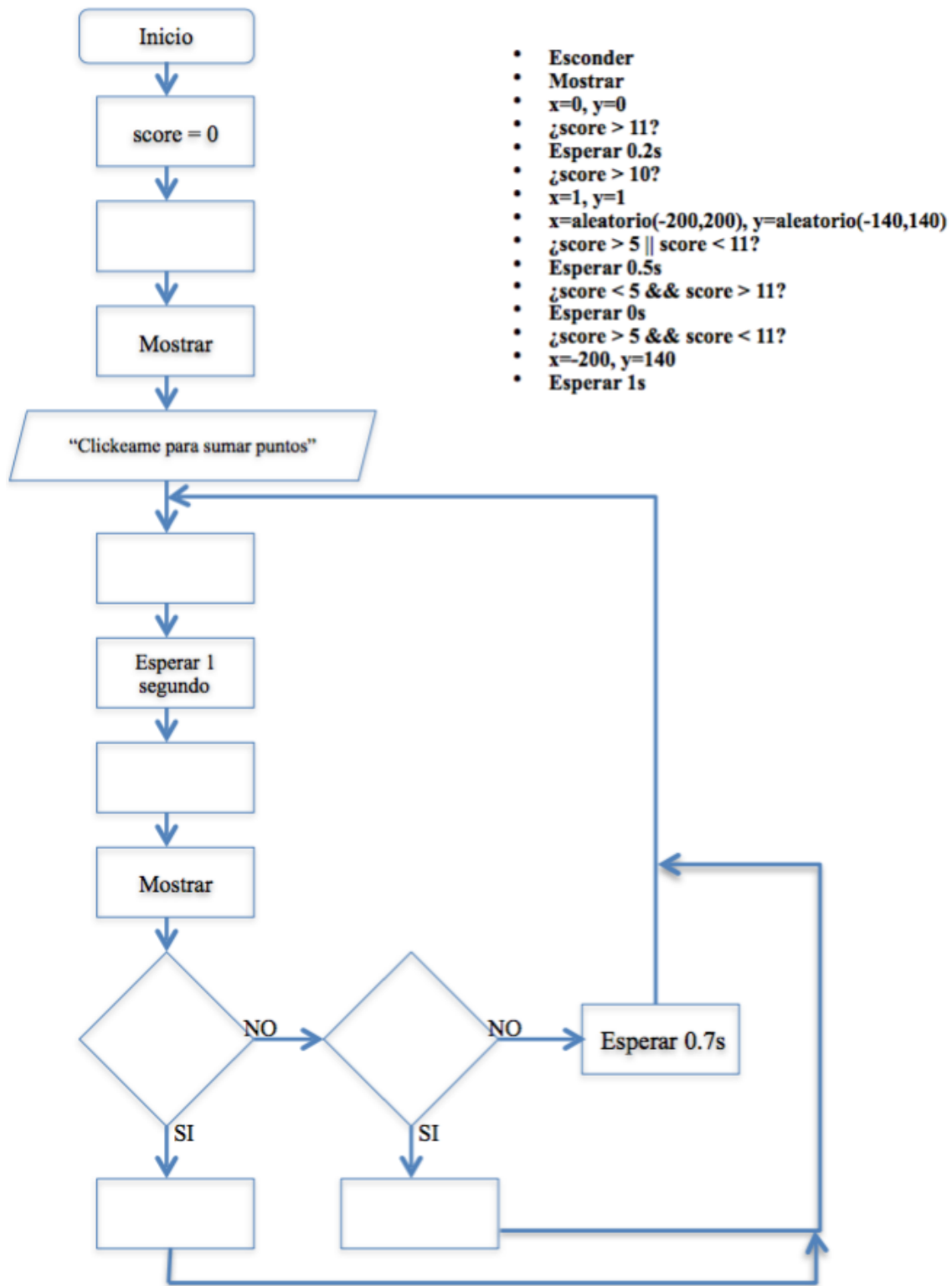
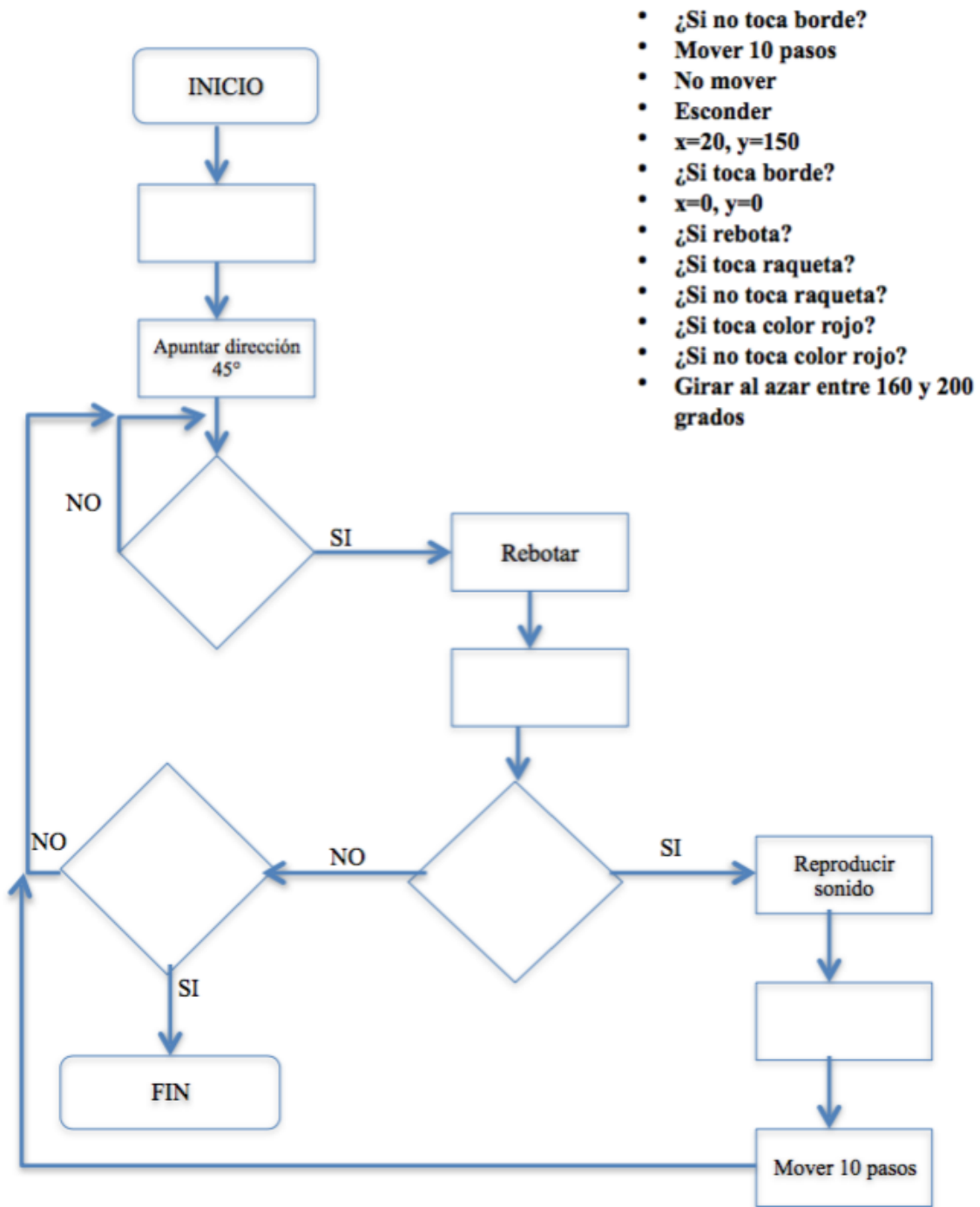


Figura 3.11: Práctica 1 segunda parte. Nivel de dificultad alto.



- ¿Si no toca borde?
- Mover 10 pasos
- No mover
- Esconder
- x=20, y=150
- ¿Si toca borde?
- x=0, y=0
- ¿Si rebota?
- ¿Si toca raqueta?
- ¿Si no toca raqueta?
- ¿Si toca color rojo?
- ¿Si no toca color rojo?
- Girar al azar entre 160 y 200 grados

Figura 3.12: Práctica 2. Nivel de dificultad alto.

**XXXI Semana de Ciencias,  
Taller de animaciones con Scratch  
del 21 al 24 de octubre**

Para realizar los ejercicios, sigue las indicaciones del facilitador. Por favor proporciona la siguiente información:

Edad: \_\_\_\_\_ Grado: \_\_\_\_\_ Escuela: \_\_\_\_\_ Fecha: \_\_\_\_\_

Conocimientos de computación: si  no

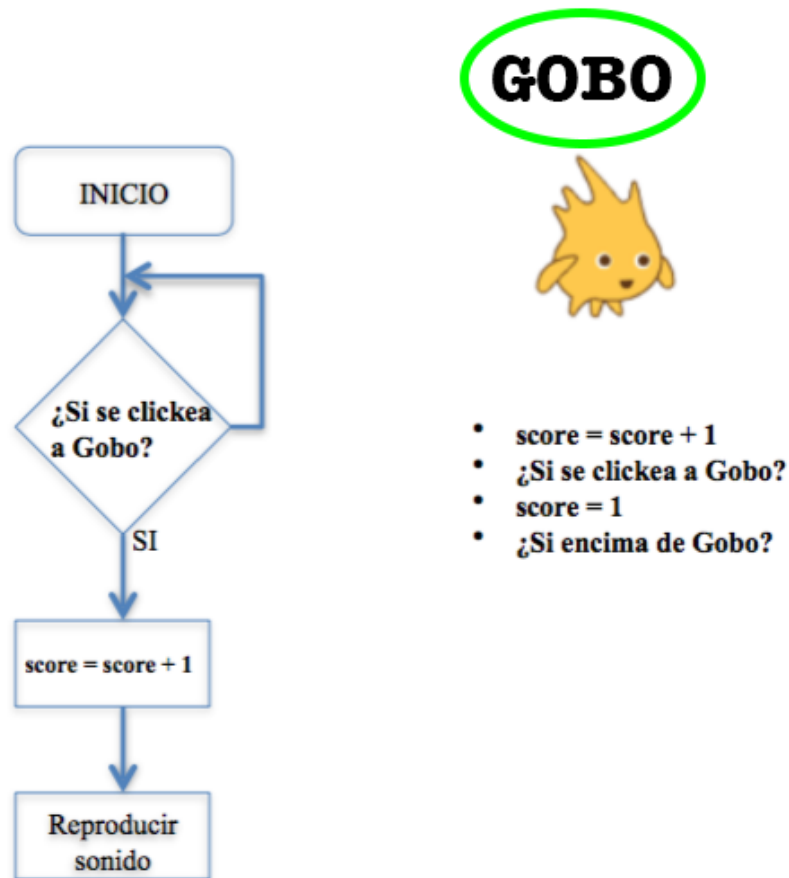


Figura 3.13: Práctica 1 primera parte. Respuestas.

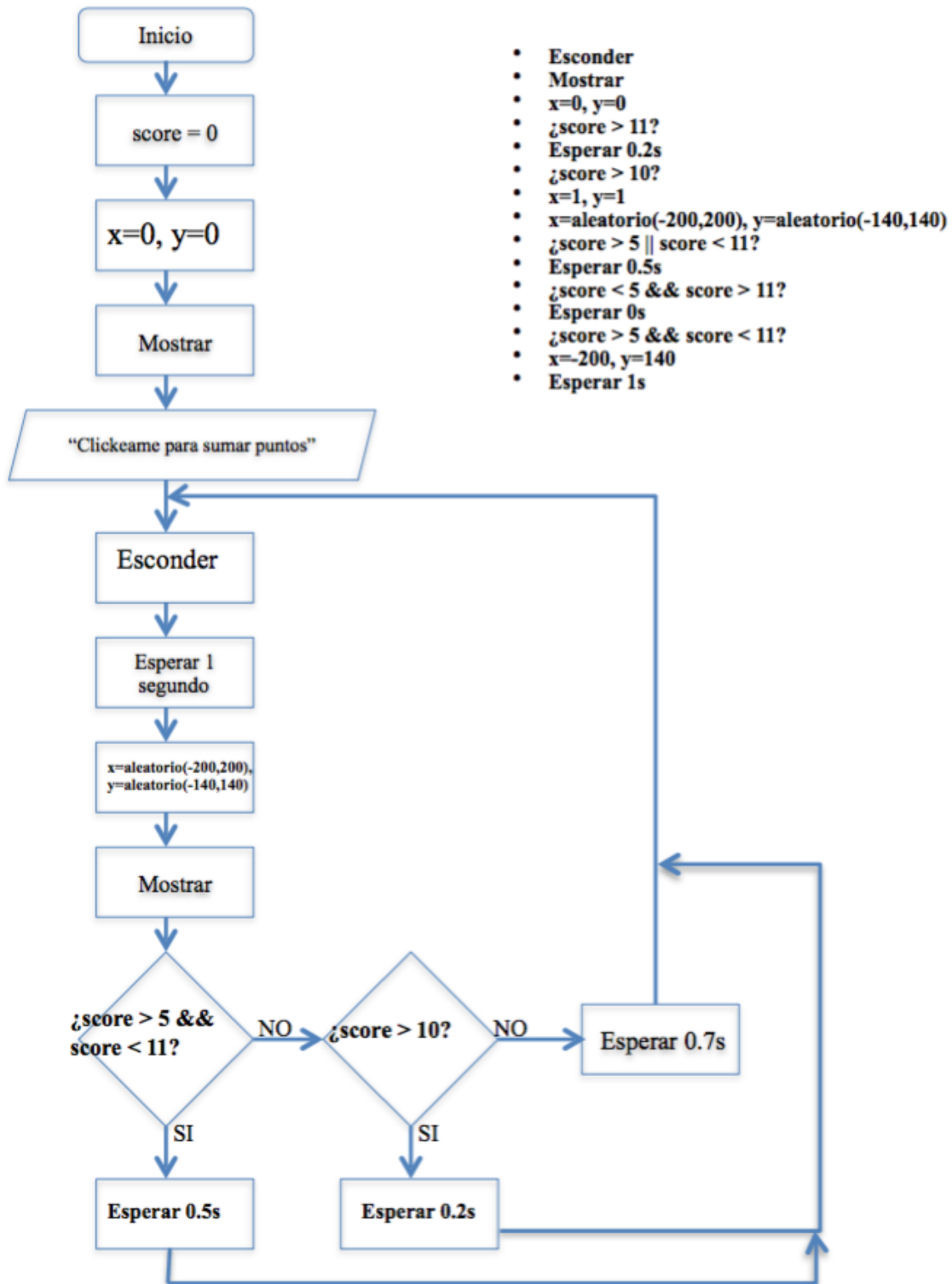
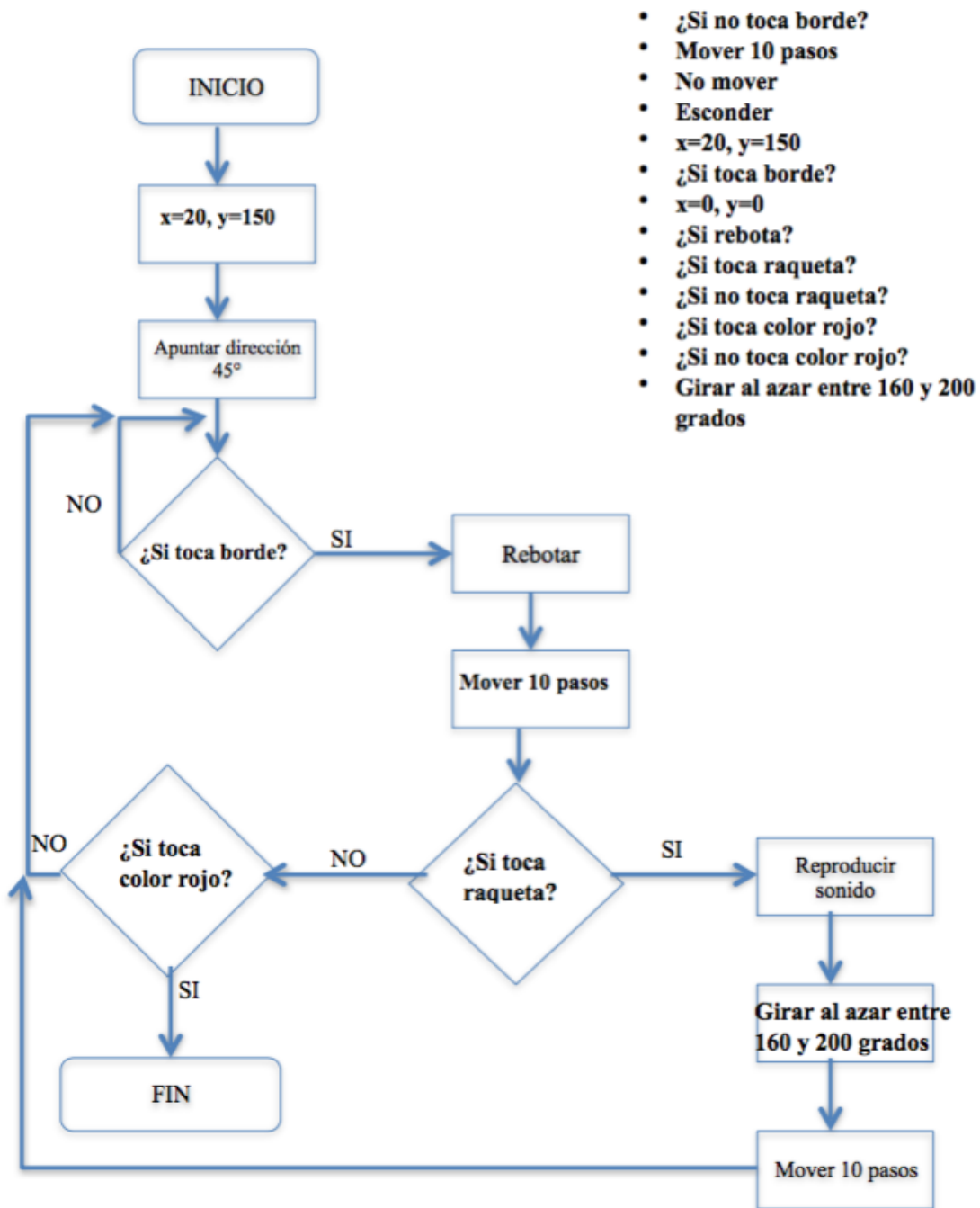


Figura 3.14: Práctica 1 segunda parte. Respuestas.



- ¿Si no toca borde?
- Mover 10 pasos
- No mover
- Esconder
- x=20, y=150
- ¿Si toca borde?
- x=0, y=0
- ¿Si rebota?
- ¿Si toca raqueta?
- ¿Si no toca raqueta?
- ¿Si toca color rojo?
- ¿Si no toca color rojo?
- Girar al azar entre 160 y 200 grados

Figura 3.15: Práctica 2. Respuestas.

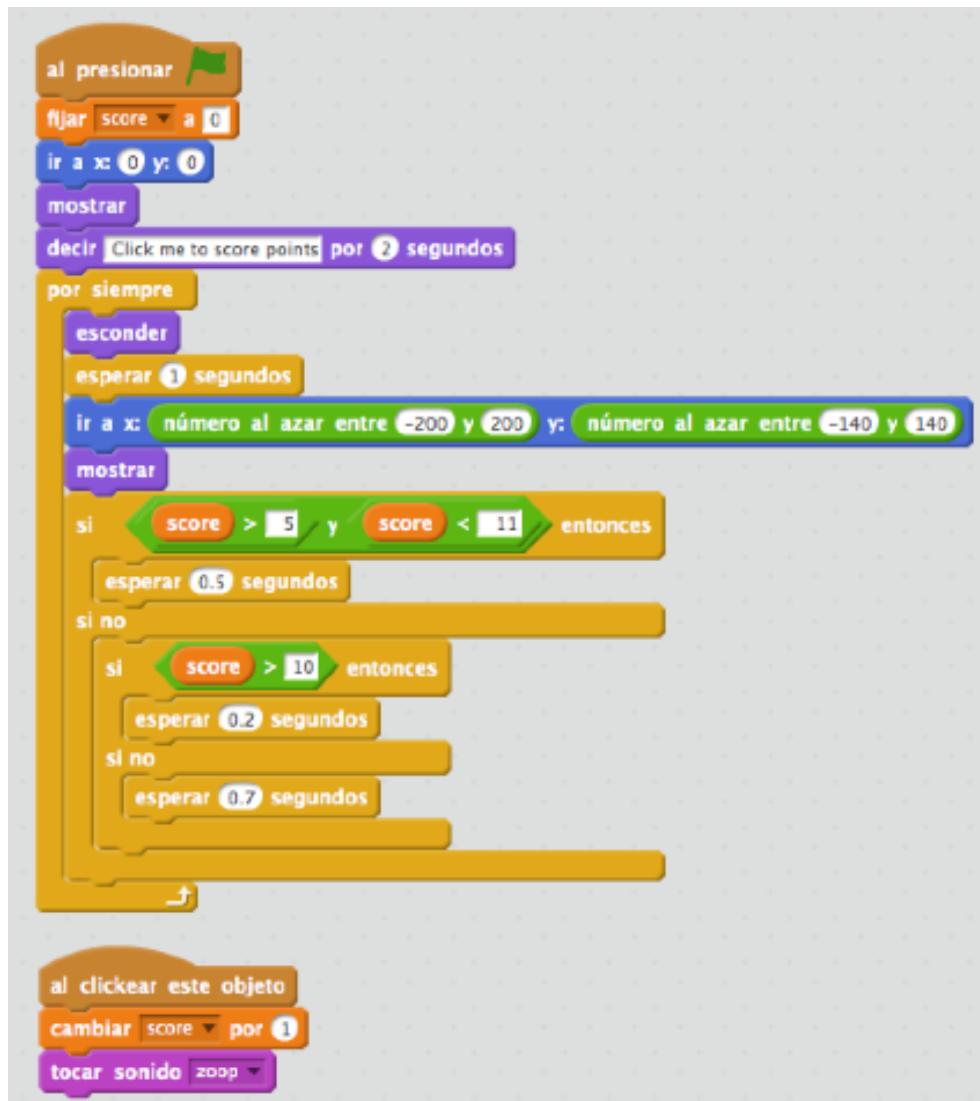


Figura 3.16: Práctica “Guaca Gobo”.

- Práctica 2.** Esta práctica tiene un alto nivel de dificultad ya que solo se describe la práctica y no se les da indicaciones para completar el diagrama a diferencia de la práctica anterior. En esta práctica se desarrollará el juego de Pong. Debemos de dibujar un círculo que representa la pelota, una semi-recta que representa la raqueta y una franja roja que ocupa el ancho del fondo que nos indicará que hemos perdido al dejar caer la pelota. Este juego consiste en que cada vez que la pelota toque a la raqueta está debe rebotar. El juego se acaba cuando la pelota llega a tocar la zona roja.



Figura 3.17: Código pong en Scratch.



Figura 3.18: Código de la raqueta en Scratch.

### 3.3.1. Objetivo

Con esta actividad se buscó que los alumnos adquieran habilidades en diseñar programas utilizando diagramas de flujo. Además la destreza de programar usando un lenguaje amigable e intuitivo como lo es Scratch. Las prácticas que se diseñaron para los alumnos fueron basados en juegos, con el objetivo que fueran divertidas y que a su vez estuvieran aprendiendo. Esta actividad se centró en el desempeño académico o conocimiento previo de los participantes, por lo que para poder medirlo se diseñaron las prácticas en papel.

### 3.3.2. Resultados

La actividad se llevó a cabo con los asistentes al taller de “Animaciones con Scratch”, que se impartió en la Semana de Ciencias de la Facultad de Ciencias. Se emplearon dos prácticas de diferente nivel de dificultad. En el taller participaron 20 alumnos de secundaria y 35 de dos preparatorias distintas (13 de Preparatoria 1 y 22 de Preparatoria 2). Se realizó una sesión de una hora de duración con cada grupo. A continuación se muestra los resultados obtenidos (figura 3.19).

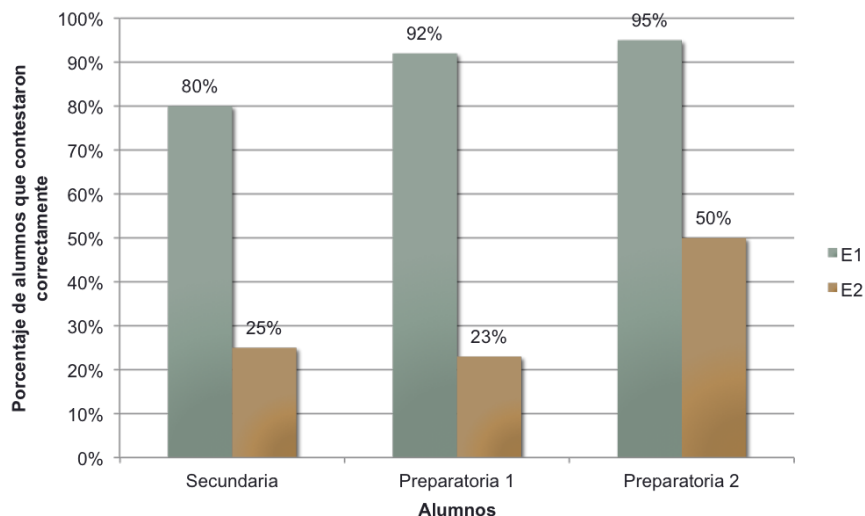


Figura 3.19: Resultados XXXI Semana de Ciencias.

Como se puede observar en la gráfica, el nivel de dificultad cambia dramáticamente entre las prácticas, se percibió una diferencia del 55 % en los alumnos de secundaria, y un 69 % y 45 % para los alumnos de preparatoria. Con base en las diferencias tan marcadas observadas entre un ejercicio y otro, la importancia de generar ejercicios con incremento paulatinos en su grado de dificultad es necesario para que el ejercicio se un reto para el alumno sin llegar a frustrarlo por la dificultad, llevando a que pierda el interés de intentarlo.

### 3.3.3. Conclusión

Gracias a los resultados obtenidos en la evaluación se concluye que el ambiente tiene grandes posibilidades de ayudar a los alumnos para aprender programación, ya que recibió buenos comentarios por parte de ellos, debido a que era un ambiente didáctico donde solamente se utilizaba lapiz y papel, y un lenguaje con interfaz gráfica como lo es Scratch, por que al no utilizar un lenguaje de alto nivel evita generar otros tipos de dificultades que complican evaluar la lógica de los alumnos. Sin embargo, se observó que hay una gran diferencia de nivel de complejidad entre las prácticas que se diseñaron, por lo que se determinó la importancia de que las prácticas vayan aumentando el nivel de complejidad paulatinamente.

## 3.4. Actividad 3: Programación en un ambiente web

Se desarrolló un ambiente web para manipular diagramas de flujo, mediante el cual el estudiante puede diseñar algoritmos para realizar prácticas de animaciones con Scratch. Las prácticas que se utilizaron en esta versión son las mismas que se usaron en la versión

de papel pero el nivel de dificultad entre cada práctica disminuyo y ahora las respuestas de los alumnos se guardan en una base de datos para su análisis posterior. Los diagramas de flujo de cada práctica están hechos con imágenes y estos están almacenados en un servidor web. Como se puede observar en la figura 3.20, el ambiente de trabajo consiste en completar mediante “Drag & Drop” un diagrama de flujo, donde se le muestran al alumno las opciones disponibles para elaborar el diagrama; solo requiere seleccionar la opción y arrastrarla al lugar donde se desea colocarla.

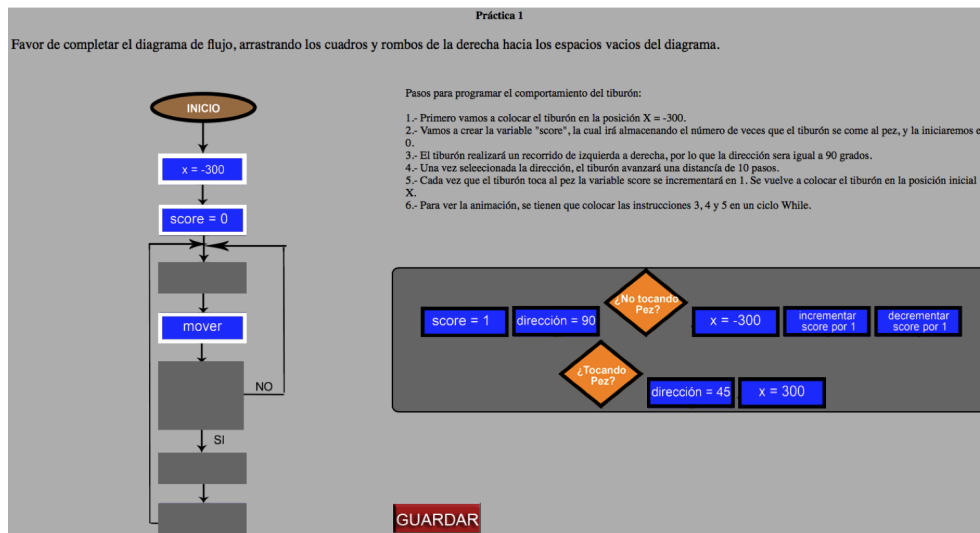


Figura 3.20: Ambiente de trabajo primera versión. Interfaz del alumno.

### 3.4.1. Objetivo

En esta actividad se enfocó en medir el conocimiento adquirido de los alumnos, por lo que se diseñaron tres prácticas a utilizar en el ambiente, también se midió la aceptación del nuevo ambiente web y la motivación que les generaba al ir realizando los ejercicios.

### 3.4.2. Resultados

En esta actividad se seleccionó al azar a algunos participantes, a quienes se les hizo una entrevista para obtener los puntos de vista de cada participante sobre la actividad, los ejercicios y el ambiente que usaron.

En esta actividad se trabajó con un grupo de 22 alumnos de cuarto semestre del Colegio de Bachilleres del Estado de Baja California (COBACH), estos alumnos tenían en su plan curricular la materia de informática donde uno de los temas es “Diseño de algoritmos”.

Los estudiantes participantes en el estudio tuvieron que completar 3 diagramas de flujo con diferente nivel de dificultad. En este experimento los alumnos realizaron las actividades en dos sesiones de dos horas. El experimento trataba de completar los diagramas al entrar a la clase (pre-test) figura 3.20, después se les daba una clase sobre el tema y tenían que volver a resolver los diagramas (post-test) figura 3.20. El primer ejercicio se trató de una pequeña animación donde tenían que programar el movimiento aleatorio de un pez y un tiburón. Los otros dos de ejercicios que se utilizaron son los mismos que se usaron en la actividad 2 “Diagramas de flujo y programación con Scratch” pero en versión web. En la siguiente figura se muestra los resultados obtenidos. Como observación, en el post-test solamente se obtuvieron 9 soluciones del ejercicio 1, 8 del ejercicio 2 y 7 del ejercicio 3 (Figura 3.21).

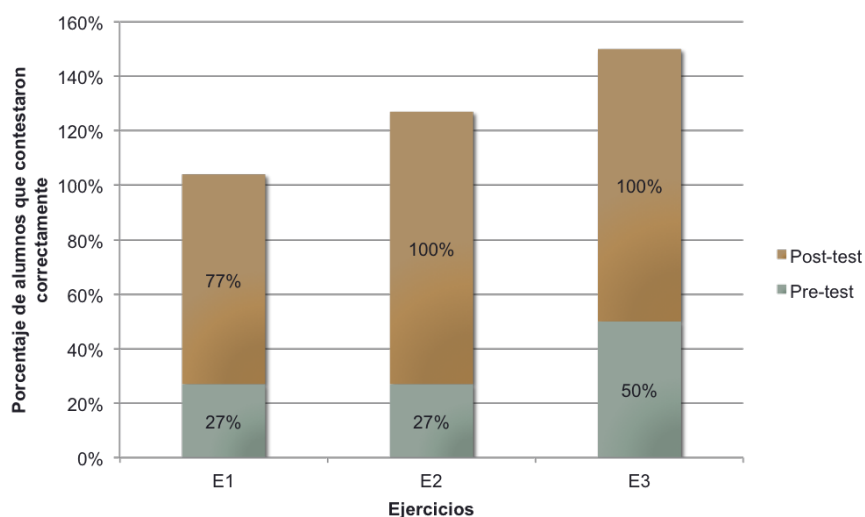


Figura 3.21: Resultados experimento COBACH.

Al final del experimento se les realizó una entrevista a algunos de los participantes donde recibimos comentarios positivos como por ejemplo “Se me hizo entretenido, se podría utilizar como apoyo para otras clases”, pero también recibimos comentarios negativos, “En las instrucciones con muchos datos se me hizo complicado pasarlo al diagrama de flujo”.

Gracias al experimento realizado y a lo comentado por los participantes se optó por diseñar prácticas con un nivel de dificultad menor e ir incrementándolo con pocas variaciones entre cada práctica. Así mismo, encontramos un problema en este ambiente de aprendizaje, para cada práctica se tenía que cortar el diagrama en imágenes, por lo que esto consumía mucho espacio en el servidor, y al momento de ingresar a la página el tiempo de carga era demasiado para mostrar el diagrama completo. Viendo este problema se decidió en realizar una segunda versión del ambiente web.

### 3.4.3. Conclusión

Con este tipo de ambiente se observó que los alumnos se interesan más en realizar las prácticas dado que ellos tomaban el completar un diagrama como un reto. Sin embargo, con relación a los aspectos técnicos del ambiente desarrollado, la forma de guardar las prácticas en el servidor no fue la más adecuada, ya que se componían de varias imágenes y esto llevó a dos cosas, que cada práctica ocupó más espacio de memoria en el servidor y que algunas veces se tardará en mostrar el diagrama. Mientras que en los aspectos didácticos se concluyó que los ejercicios deben de ser más limpios, no incluir muchos datos ya que puede llegar a confundir al alumno y perder el flujo del programa.

# Capítulo 4

## Diseño e implementación del LTRP

Para conducir nuestra investigación, se desarrolló un Laboratorio Teleoperado de Robótica Pedagógica (LTRP) que permite manipular remotamente robots LEGO Mindstorm. Para la implementación del laboratorio, se consideraron cuatro elementos principales:

1. Un servidor Web que permite la comunicación de usuarios del LTRP con robots pedagógicos, a través de un navegador Web.
2. Robots LEGO que se comunican con el servidor WEB a través de WI-FI o Bluetooth.
3. Aplicación cliente que permite el envío de las instrucciones que el usuario desee sean ejecutadas por el robot.
4. una cámara Web, para la transmisión de vídeo en tiempo real de las acciones que el robot está realizando, permitiéndole al usuario observar el comportamiento del robot al recibir las instrucciones en forma remota.

Para la cámara, también se contempla una comunicación con el servidor vía WI-FI o Bluetooth. Un ejemplo de la arquitectura propuesta se presenta en la figura 4.1.

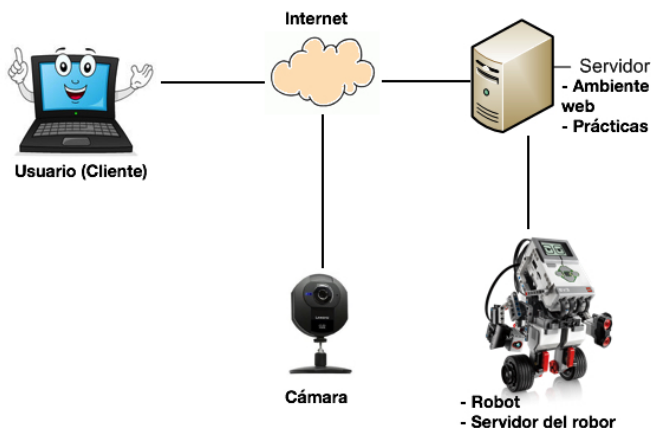


Figura 4.1: Arquitectura propuesta.

En este capítulo se presenta el diseño e implementación del LTRP dividido en dos partes, la primera parte considera el hardware, es decir todo el equipo que se utilizó, así como la conexión y comunicación del mismo. La segunda es el software, donde se explica que se desarrolló, los ambientes de programación del alumno y del maestro, las librerías que se utilizaron para habilitar un servidor dentro de la computadora del robot, y también los intérpretes que se desarrollaron para poder realizar la comunicación entre lenguajes, tales como: HTML-XML, XML-HTML y HTML-JAVA. También dentro de este capítulo se presenta el diseño e implementación del experimento realizado con alumnos de la Facultad de Ciencias de la UABC, así como los resultados obtenidos.

## 4.1. Componentes de Hardware

En esta sección se describen todos los dispositivos que se utilizaron para construir el LTRP, además se explica como se realizó la conexión de las cámaras al router y como poder transmitir en vivo las acciones del robot, así como el armado del robot, los sensores con los que estará equipado y el forma del robot.

### 4.1.1. Infraestructura

En el laboratorio de prácticas avanzadas (LPA2), perteneciente a la Facultad de Ciencias, se cuenta con el siguiente equipo que permitirá el desarrollo del trabajo propuesto:

- 1 Mesa configurable para el medio ambiente de los robots.
- 1 estación de trabajo DELL T3500.

- 2 cámaras inalámbricas, modelo WVC80N, marca CISCO. Capturan audio y vídeo, y cuentan con detección de movimiento.
- 2 robots lego Mindstorm EV3.
- 1 servidor web, Dell PowerEdge R420, para la instalación de los servicios requeridos.

#### 4.1.2. Conexión de cámaras

Para la transmisión en vivo, se utilizaron las dos cámaras de vídeo disponibles. Cada una de las cámaras están conectadas directamente al router de la red inalámbrica del laboratorio (LPA2), asignándoles una dirección IP para poder acceder a la transmisión del vídeo. En la figura 4.4 se muestra la colocación de una de las cámaras en el LTRP y como el alumno puede observar los movimientos del robot en la computadora utilizando las cámaras.



(a) Colocación de cámaras.



(b) Visualización en el ambiente.

Figura 4.2: Conexión de cámaras.

### 4.1.3. Armado del robot

En este proyecto se decidió armar un robot sencillo de LEGO Mindstorm EV3, buscando que en éste hubiera más de un sensor para trabajar. En este caso se decidió armar un robot en forma de carro, y para que no abarcará tanto espacio en la mesa solamente se le incorporaron dos sensores, que son el sensor de color y el sensor ultrasónico. Gracias a que la mesa donde se va a encontrar el robot es configurable, se pueden incrustar barreras en ella para poder trabajar con el sensor ultrasónico y esto lleva a que se puedan diseñar prácticas relacionadas con laberintos. Y para trabajar con el sensor de color se diseño una pista con diferentes colores, dándoles a los maestros la posibilidad de recorrer distintas rutas al momento de diseñar las prácticas (Observar figura 4.3).

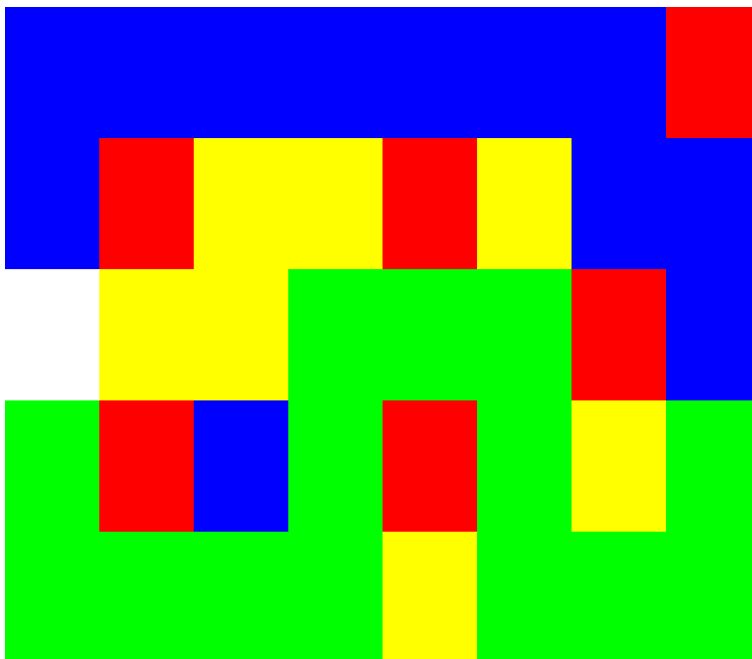


Figura 4.3: Mundo del robot.

#### 4.1.4. Evidencia fotográfica

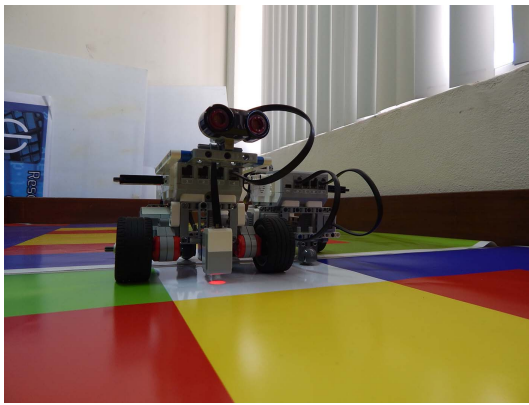
A continuación se muestra una serie de imágenes que muestran la versión final del LTRP, con los robots y las cámaras que se utilizaron.



(a) Entorno del robot y cámara.



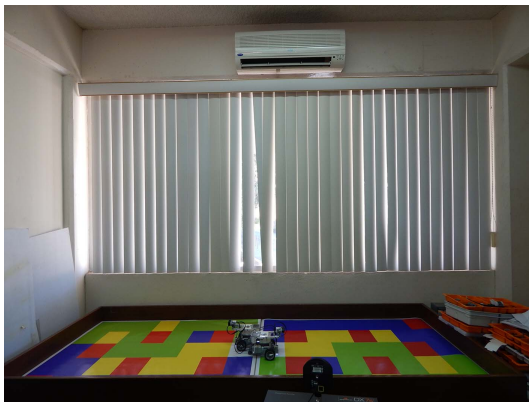
(b) Laboratorio de practicas avanzadas II.



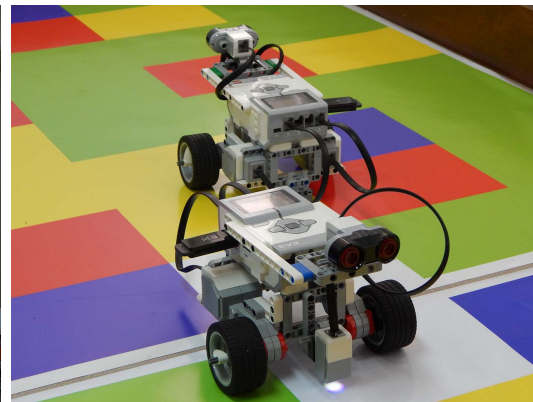
(c) Robot pedagógico I.



(d) Cámara.



(e) Entorno del robot.



(f) Robots pedagógicos II.

Figura 4.4: Laboratorio Teleoperado de Robótica Pedagógica.

## 4.2. Componentes de Software

En esta sección se explican los ambientes de programación que se diseñaron y desarrollaron para generar prácticas por parte del maestro y resolver dichas prácticas por

parte de los alumnos. Además, se aborda el programa que utiliza el robot para realizar las acciones que se le envían por medio del ambiente del alumno.

### 4.2.1. Ambiente de programación

En un escenario típico al utilizar el LTRP, el alumno seleccionará, de una lista de temas disponible, la temática de programación donde desee reforzar su conocimiento (ej. condiciones, ciclos, funciones, entre otros); indicando el grado de complejidad deseado. El alumno recibirá las indicaciones para realizar el ejercicio, el cual tendrá como objetivo programar una serie de acciones que le permitan al robot realizar una tarea; por ejemplo, trasladarse de un punto a otro, evadiendo una serie de obstáculos, etc.

El estudiante realizará un programa, consistente de una serie de comandos, buscando que el robot realice la tarea indicada, el programa será transmitido al robot y el estudiante podrá visualizar el comportamiento del robot mediante la transmisión de vídeo.

El LTRP se divide en dos partes, la primera parte es el ambiente de programación para el alumno y el maestro, la segunda parte es el programa del robot. El ambiente de programación que se desarrolló para el LTRP se decidió que fuera web, ya que la idea que se tiene es que los alumnos pueden utilizar el ambiente desde su casa solamente contando con una conexión a Internet.

#### 4.2.1.1. Ambiente del maestro

El ambiente del maestro se desarrolló para realizar ejercicios para el laboratorio de una forma amigable y fácil de utilizar. Mediante “Drag & Drop” el maestro va seleccionando y arrastrando los elementos que quiere incluir en el diagrama, así como las opciones que tiene el alumno por completar el diagrama. Como se puede observar en la figura 4.5, el ambiente cuenta con tres menús, el primero es para configurar las acciones que puede realizar el robot (figura 4.6), el segundo es donde se encuentran las estructuras de control (figura 4.7), y el tercero es donde se asignan las variables de los sensores conectados al robot, esta opción se agregó gracias a los comentarios recibidos por los maestros que utilizaron el ambiente descritos más adelante.



Figura 4.5: Menú principal.

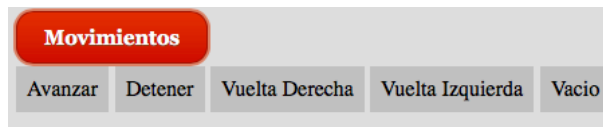


Figura 4.6: Opciones de movimiento.

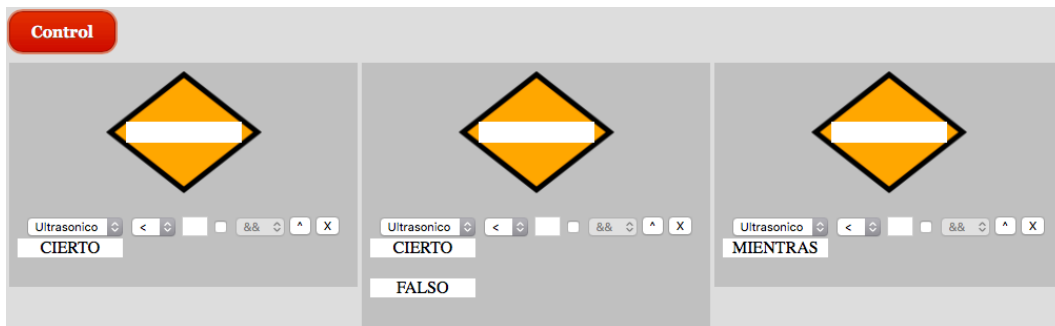


Figura 4.7: Opciones de estructuras de control.

El maestro selecciona las opciones que le da el ambiente para generar el diagrama de flujo que quiere que conteste el alumno, proporcionándole también las opciones que tiene para completarlo. Una vez que está conforme con lo seleccionado, como en el ejemplo mostrado en la figura 4.8, el ejercicio se guarda como archivo XML (figura 4.9) en el servidor, con esto ahorramos espacio de almacenamiento y los tiempos de carga para cada ejercicio es más rápido. En la figura 4.10 se muestra el flujo para generar las prácticas por parte del maestro hasta guardarse en el servidor. Para generar el archivo XML se creó un intérprete donde se reciben los datos en HTML y una por una se van creando las etiquetas en XML con identificadores y nivel de profundidad, que nos sirve para saber que elemento pertenece a cada estructura de control (figura 4.11).

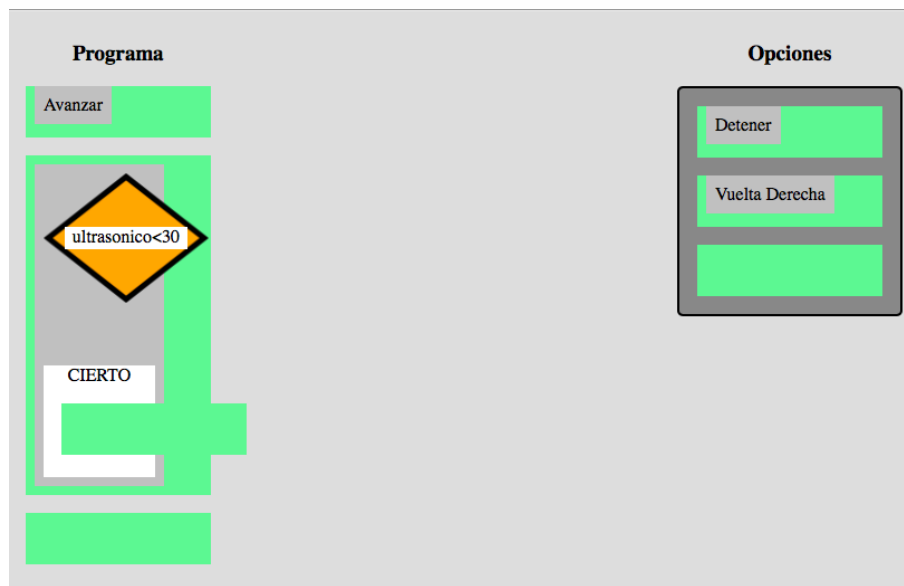


Figura 4.8: Ejemplo como ir creando un ejercicio.

```
ejercicio1.xml
1 <?xml version="1.0"?>
2 <contenido>
3   <diagrama id="1" class="while-def">
4     <imagen>vacio</imagen>
5   </diagrama>
6   <diagrama id="2" class="vacio-def">
7     <imagen>vacio</imagen>
8   </diagrama>
9   <opciones id="1" class="condicional">
10    <imagen>color*!=*rojo</imagen>
11  </opciones>
12  <opciones id="1" class="accion">
13    <imagen>Avanzar</imagen>
14  </opciones>
15 </contenido>
16
```

Figura 4.9: Ejemplo del archivo XML correspondiente al ejercicio de la figura 4.8.

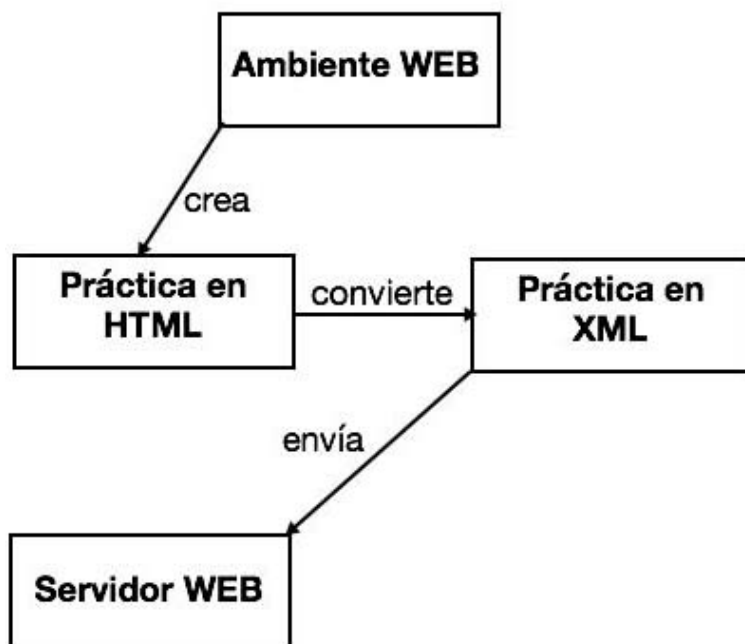


Figura 4.10: Flujo para crear práctica en XML.

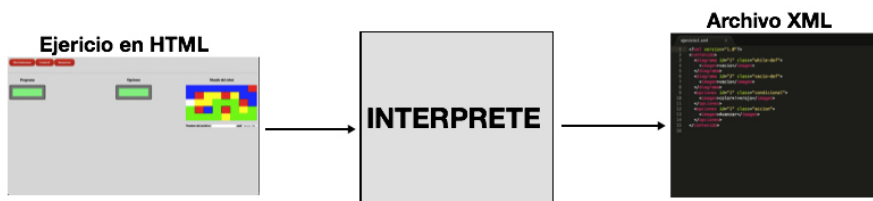


Figura 4.11: Diagrama: Conversión HTML - XML.

Esta ambiente se mostró a los maestros, con el motivo que lo usaran y realizaran sus observaciones para mejorarlo. Uno de los puntos que comentaron fue “Agregar una foto del entorno donde se encuentra el robot, para que el maestro tenga una idea de las prácticas que puede realizar”, “Agregar una foto del robot con los sensores que tiene incluidos, para darnos una idea con que sensores se puede trabajar”, “Ponerle títulos en las secciones que existen dentro del Framework para saber que es cada parte” y “Solamente incluir los sensores con los que va a trabajar el robot para mostrarlos al construir la condicional en el Framework”. Gracias a estos comentarios realizados por los maestros se mejoró el ambiente, para finalmente quedar como se muestra en las figuras 4.12, 4.13.

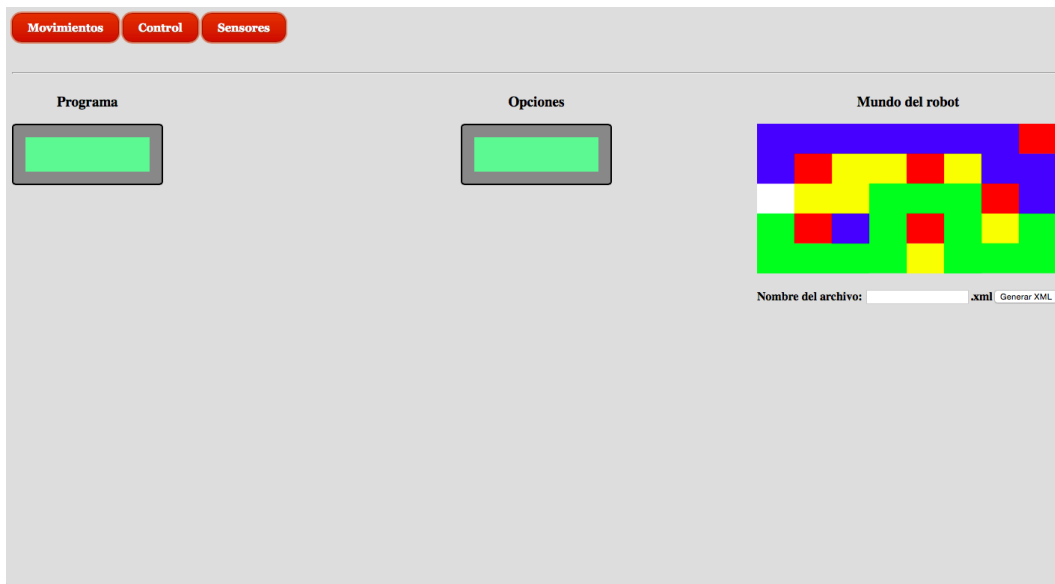


Figura 4.12: Ambiente de programación segunda versión. Interfaz del maestro

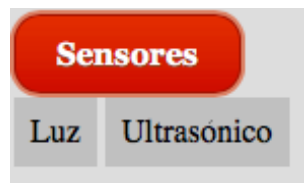


Figura 4.13: Opciones de sensores.

#### 4.2.1.2. Ambiente del alumno

Una vez que el maestro crea el ejercicio, éste se guarda en el servidor como archivo XML. Cuando el alumno ingresa a realizar el ejercicio, se realiza la conversión de XML a elementos de HTML, lo anterior permite mostrar el diagrama de flujo (figura 4.14). En la figura 4.15 se muestra el flujo comenzando desde que la práctica está guardada como archivo XML dentro del servidor. Una vez que se selecciona la práctica que se va a realizar, ésta se debe de desplegar en elementos HTML para que el alumno pueda observar el diagrama y así poder completarlo. Para realizar esta tarea se desarrolló un intérprete, donde se reciben las etiquetas que se guardan en XML, se lee la información que contiene para después decidir si es un recuadro de acción o una estructura de control y por último despliega que acción es la que viene escrita o que tipo de estructura de control (figura 4.16).

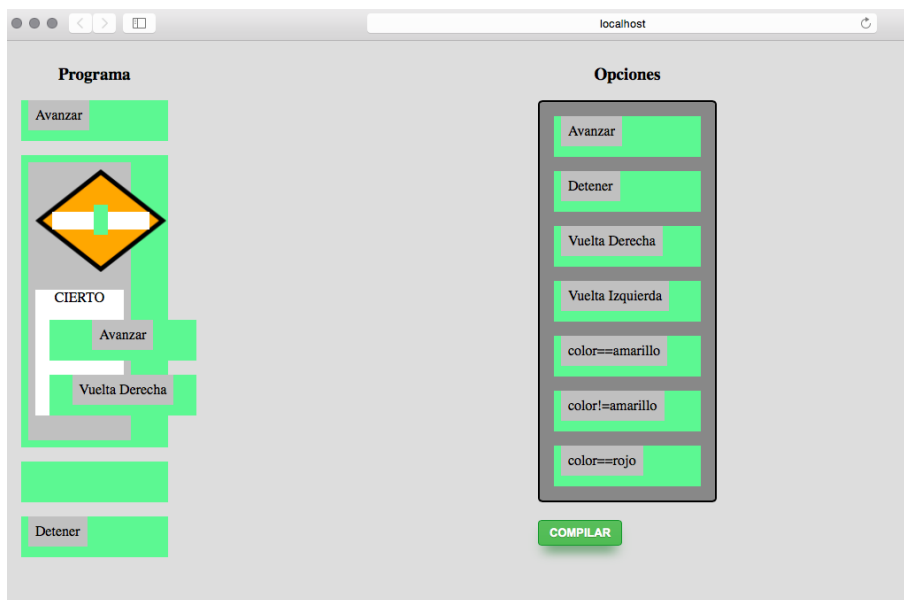


Figura 4.14: Ejemplo de ejercicio para el alumno.

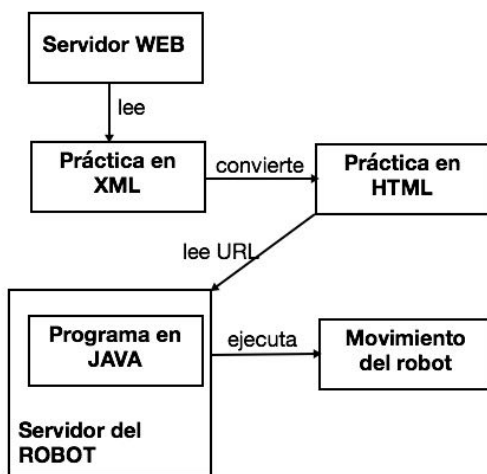


Figura 4.15: Flujo para cargar el archivo en XML en el ambiente.



Figura 4.16: Diagrama: Conversión XML - HTML.

El alumno debe de completar el diagrama de flujo en este ambiente, que son las acciones que puede realizar el robot. Una vez que el alumno termina de completar tiene que compilar su programa dándole click en el botón de compilar, este botón va a mostrar el pseudocódigo del diagrama si todo está bien o muestra un mensaje de error si el diagrama no está completo. Una vez compilado el programa ya puede ejecutarlo y observar el movimiento del robot mediante las cámaras web que están conectadas en el laboratorio.

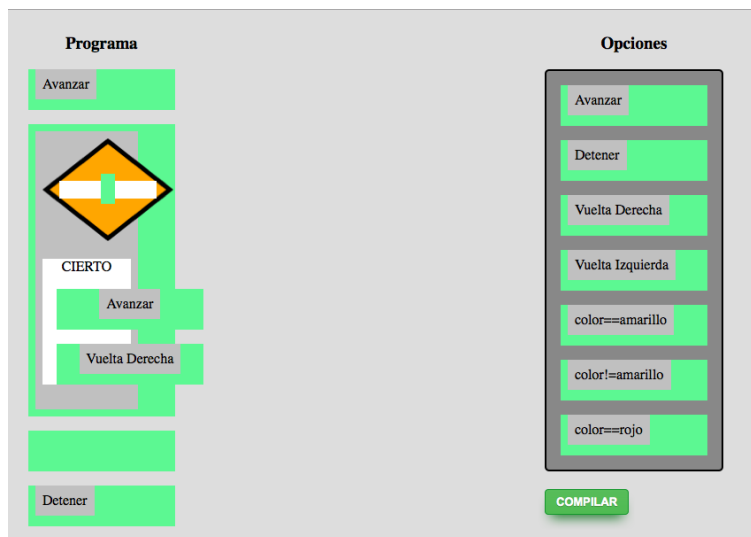


Figura 4.17: Inicio de la práctica.

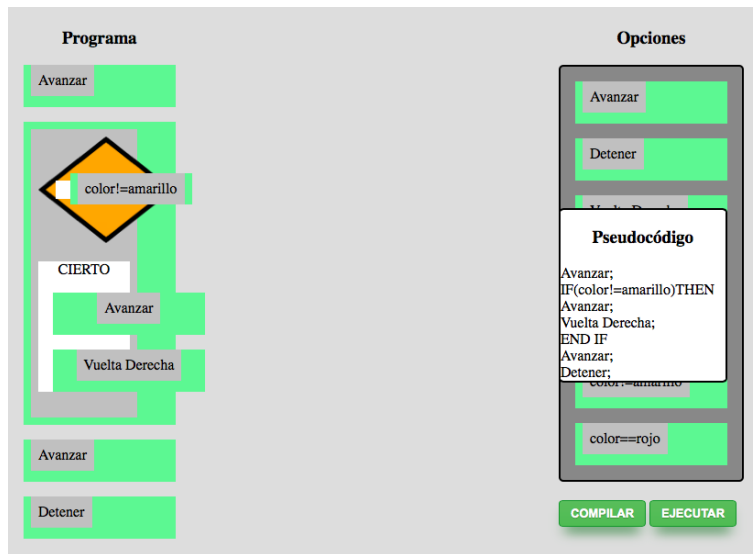


Figura 4.18: Práctica compilada y pseudocódigo.

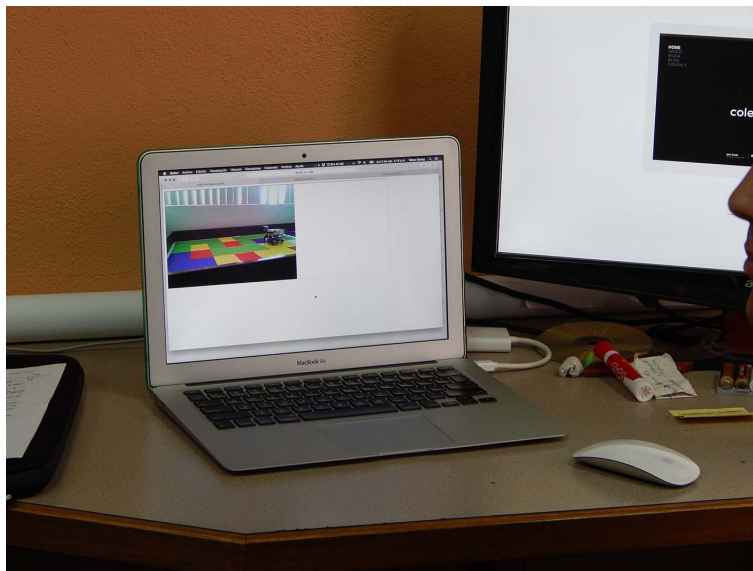


Figura 4.19: Visualización del robot I.



Figura 4.20: Visualización del robot II.

#### 4.2.1.3. Programa del robot

El siguiente paso al momento de realizar un ejercicio, y quizá el mas importante, es el programa del robot y el servidor del robot. Primero hablaremos sobre el servidor del robot, en vez de realizar una conexión del robot al servidor para poder utilizarlo de forma remota, se decidió instalar un servidor web dentro de la memoria del robot, que tuviera las mismas características que un servidor normal pero para dispositivos de baja gama. El tipo de servidor que se utilizó es llamado nanoHTTPD, considerado como un servidor “ligero”, es decir es un servidor que no requiere muchos recursos para un funcionamiento estable. Este tipo de servidores están enfocados a estar embebidos en dispositivos con poca memoria de procesamiento y almacenaje, así como en aplicaciones. El objetivo es que el dispositivo o la aplicación tenga su propio servidor HTTP para permitir la comunicación a través de la red. Una vez que se ya tiene el servidor instalado le asignamos una IP a cada robot para realizar la comunicación, es aquí donde se realiza la segunda parte del programa del robot, que recibe las acciones que se completaron en el diagrama de flujo y poder realizar los movimientos. Hay que destacar que la información de las acciones y estructuras de control que tiene el diagrama, se guardaron en un formato string o cadena, por lo tanto dentro del robot se creó un analizador léxico, donde separa la cadena por caracteres, después analiza cada carácter y decide que acción va a realizar o que estructura corresponde con su condición.

# Capítulo 5

## Evaluación del LTRP

Las preguntas de investigación dos y tres, ¿Cuáles son las características que deben ser consideradas al elaborar las prácticas interactivas que mantengan el interés del alumno y refuerzan el aprendizaje de los temas? y ¿Qué elementos de las metodologías de diseño instruccional deben ser considerados para la elaboración de las prácticas interactivas en el LTRP?, se contestaron con base en el resultado de un estudio donde profesores y alumnos participantes, realizaron una evaluación formativa y sumativa sobre el impacto del LTRP y el material instruccional desarrollado. Lo anterior nos permitió elaborar una guía de diseño e implementación de ejercicios utilizando laboratorios teleoperado. A continuación se describen a detalle cada una de las evaluaciones formativas y la evaluación sumativa con los diferentes ambientes que se desarrollaron.

### 5.1. Métodos utilizados en el estudio

Para medir el desempeño académico de los estudiantes, así como la aceptación en el uso del LTRP, aspectos de mayor interés en esta investigación, se consideró la siguiente pregunta de investigación: 1) ¿Cuáles son las características que deben ser consideradas al elaborar las prácticas interactivas que mantengan el interés del alumno y refuercen el aprendizaje de los temas? Para dar respuesta a esta pregunta, se definieron las siguientes componentes en la metodología utilizada en el estudio.

### 5.2. Participantes

En la Facultad de Ciencias de la Universidad Autónoma de Baja California se seleccionó un grupo de tronco común que llevan la materia de “Introducción a la Programación”. A este grupo se les aplicó un examen de diagnóstico sobre el tema de estructuras de control, que nos sirvió como indicador para seleccionar a los alumnos que manifestaron mayor deficiencia en el tema, de los cuales se seleccionaron 10 alumnos que formaron el grupo experimental.

### 5.3. Duración del estudio

El estudio inició aplicándole a los alumnos un examen diagnóstico (En el apéndice C se muestra el examen que se utilizó tanto el pre-test como en el post-test) sobre el tema de estructuras de control, donde se busca que los alumnos cumplan con cuatro competencias que son:

- Abstracción de datos
- Secuencia
- Orden
- Lógica programática

Cada una de las preguntas expuestas en el examen de diagnóstico están basadas en medir una o varias de las competencias mencionadas anteriormente (En la tabla 5.1 se puede observar la relación que hay entre las preguntas del pre-test y post-test con las cuatro competencias). Por ejemplo, la pregunta 2 que dice “Selecciona el orden para describir como harías un sándwich (Teniendo todo lo requerido en la cocina)”, evalúa si el alumno es competente al especificar la secuencia y en el orden para generar el pseudocódigo para realizar la tarea.

Tabla 5.1: Relación preguntas-competencia

Preguntas	Competencia
Pregunta 1	Lógica programática
Pregunta 2	Abstracción de datos - Secuencia - Orden
Pregunta 3	Lógica programática
Pregunta 4	Lógica programática
Pregunta 5	Abstracción de datos - Secuencia - Orden
Pregunta 6	Lógica programática
Pregunta 7	Lógica programática
Pregunta 8	Abstracción de datos - Secuencia - Orden
Pregunta 9	Lógica programática
Pregunta 10	Abstracción de datos - Secuencia - Orden
Pregunta 11	Abstracción de datos - Secuencia - Orden - Lógica programática

Es importante hacer hincapié, que antes de realizar el examen de diagnóstico, los alumnos ya habían recibido información sobre el tema por parte del docente. El grupo experimental utilizó el LTRP para desarrollar dos prácticas, en la figura 5.1 podemos observar un ejemplo de estas, sobre el tema de estructuras de control. Dentro de las estructuras de control se encuentran dos tipos: Selectivas e Iterativas. En el LTRP se trabajó con los alumnos las estructuras de control selectivas, aquellas donde se utilizan “IF-THEN” y “IF-THEN-ELSE”, en ellas se utilizaron condiciones simples y múltiples.

Al inicio del experimento a los estudiantes se les dio instrucciones sobre lo que se iba a realizar, después prosiguieron a trabajar con la primera práctica donde tenía que utilizar una estructura de control selectiva. En la segunda práctica siguieron trabajando con estructuras de control selectivas, pero esta vez tenía que llevar una condición múltiple. Una vez que terminaron las prácticas se les entregaba su examen de diagnóstico previamente contestado, para que corrigieran los errores que habían obtenido, una vez que se corrigieron las respuestas se les dio retroalimentación sobre el tema para concretar lo aprendido. Al final del experimento se condujo con cada uno de los alumnos participantes una encuesta para obtener retroalimentación cualitativa del uso del LTRP. A continuación se muestran las preguntas que se les hizo en la entrevista:

- ¿Se te hizo fácil utilizar el ambiente?
- ¿Consideras que te ayudó a comprender los temas?
- ¿Crees que te ayudaría en otros temas?
- ¿Te gustaría usar el LTRP desde tu casa?
- ¿Algún comentario o sugerencia sobre el LTRP?

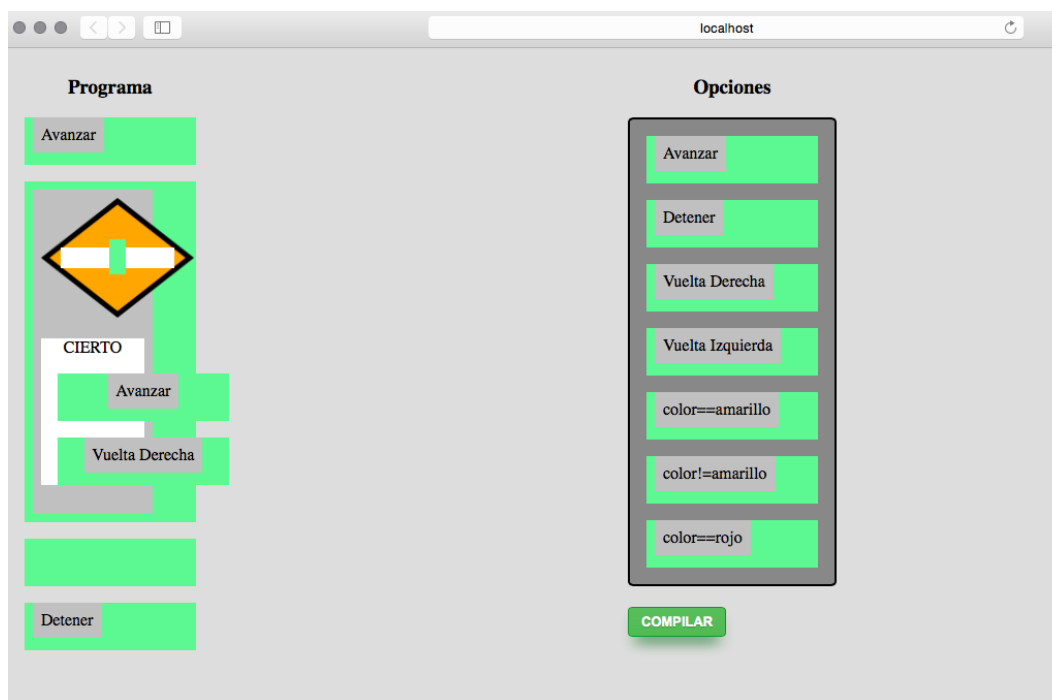


Figura 5.1: Ejemplo de práctica del LTRP.

## 5.4. Fuente de datos y mecanismos de análisis

Para dar respuesta a la pregunta de investigación planteada, utilizamos datos de dos fuentes distintas:

- I . Calificaciones finales del post-test.
- II . Experiencia de uso expresada por los alumnos que utilizaron el sistema (Encuesta y Vídeo).

Los datos antes mencionados se utilizaron como se describe a continuación. Primero, para la evaluación del impacto académico y desempeño educativo de los alumnos que utilizaron el LTRP, se consideró la fuente sobre las calificaciones finales del post-test. Posteriormente , en la evaluación del grado de aceptación y el nivel de satisfacción del alumno que usaron el LTRP, utilizamos la fuente de experiencia de uso del sistema.

## 5.5. Resultados del estudio

Los resultados que se obtuvieron de este estudio preliminar no son representativos si hablamos del aprovechamiento ya que para poder considerar el desempeño del ambiente desarrollado se tendría que incorporar el LTRP al plan de trabajo del docente e ir observando a los alumnos que lo utilizan en el semestre. Pero a su vez se obtuvo un resultado importante en la motivación que se da al alumno para seguir practicando con el robot. A continuación, se describen los detalles de los resultados obtenidos.

### 5.5.1. Beneficios obtenidos en el nivel de aprovechamiento de los alumnos

El tiempo invertido por los alumnos, quienes voluntariamente utilizaron el LTRP, y el número de participantes se consideró representativo para el estudio preliminar que se llevó a cabo. A pesar de que la comparación entre las calificaciones del pre-test y el post-test son favorables (tabla 5.2), se considera que es necesario un mayor número de participantes y de tiempo de uso del LTRP, para concluir que el laboratorio impacta positivamente en el desempeño educativo de los alumnos.

Tabla 5.2: Resultados Pre-test y Post-test.

Participante	Calificación pre-test	Calificación post-test
1	5.5	8.2
2	5.5	9.1
3	4.5	9.1
4	4.5	8.2
5	5.5	10
6	4.5	8.2
7	3.6	8.2
8	4.5	10
9	2.7	8.2
10	2.7	9.1
Promedio general	4.35	8.83

### 5.5.2. Aceptación en el uso del ambiente de aprendizaje

Las preguntas de la encuesta se categorizaron en 3 grupos, como se muestra en la tabla 5.3. La primera categoría es la usabilidad del ambiente, la cual considera aspectos como que tan fácil es utilizar el ambiente de programación, si es agradable la interfaz gráfica, entre otros. Referente a esta categoría obtuvimos una apreciación aceptable. El 65 % de los entrevistados manifestaron una experiencia positiva. Sin embargo se obtuvieron comentarios encontrados, por ejemplo un participante nos comentó “si se hizo fácil, está muy práctico”, mientras otro participante nos comentó “Más o menos. Está muy enredoso”.

La siguiente categoría es si el LTRP ayudó al alumno a comprender el tema visto obteniendo un 90 % de satisfacción. Por ejemplo unos de los comentarios positivos fue “Si me ayudó por que veía que estoy haciendo y miré el resultado cuando el robot se movió”, pero el único comentario negativo fue “Me ayudó muy poco”.

Y por último, en la categoría sobre la motivación que brinda el LTRP para que continúen realizando prácticas se obtuvo un 100 % de aceptación. Algunos participantes comentaron “Ver como se mueve el robot es más entendible lo que hago”, “es más práctico y más divertido porque así ya miras realmente lo que estas haciendo y no solo programarlo y ya”.

Tabla 5.3: Comentarios de los entrevistados en categorías.

Categorías	Positivo	Negativo
Usabilidad del ambiente <ul style="list-style-type: none"> <li>▪ ¿Se te hizo fácil utilizar el ambiente?</li> <li>▪ ¿Algún comentario o sugerencia sobre el LTRP?</li> </ul>	65 %	35 %
Tema visto en el LTRP <ul style="list-style-type: none"> <li>▪ ¿Consideras que te ayudó a comprender los temas?</li> <li>▪ ¿Crees que te ayudaría en otros temas?</li> <li>▪ ¿Algún comentario o sugerencia sobre el LTRP?</li> </ul>	90 %	10 %
Motivación <ul style="list-style-type: none"> <li>▪ ¿Te gustaría usar el LTRP desde tu casa?</li> <li>▪ ¿Algún comentario o sugerencia sobre el LTRP?</li> </ul>	100 %	0 %

### 5.5.3. Análisis de vídeos

Al observar los vídeos que se grabaron durante el estudio, como era esperado por los investigadores, se pudo observar que al principio hubo problemas al utilizar el ambiente, algunos alumnos batallaron en familiarizarse al completar el diagrama de flujo mostrado con la primera práctica, generando gestos de frustración. Al comienzo de la segunda práctica fue más fluido el completar el diagrama de flujo, lo que llevó a que se terminara más rápido la práctica. A continuación se detalla en dos etapas, lo que se observó directamente en el estudio y en los vídeos: Ambiente de programación y Visualización del robot.

#### 5.5.3.1. Ambiente de programación

El ambiente de programación tuvo una respuesta aceptable, pero se pudo observar que al iniciar las prácticas realizaban gestos y comentarios de inconformidad, porque a primer instancia no sabían que hacer o como se utilizaba el ambiente. Pero esto cambió al iniciar la segunda práctica, ya que se habían familiarizado con el ambiente. Esto dio como resultado que el tiempo de duración entre la primera y la segunda práctica se redujo significativamente, presentando una diferencia de 15 minutos aproximadamente. (Observar figuras 5.2, 5.3).

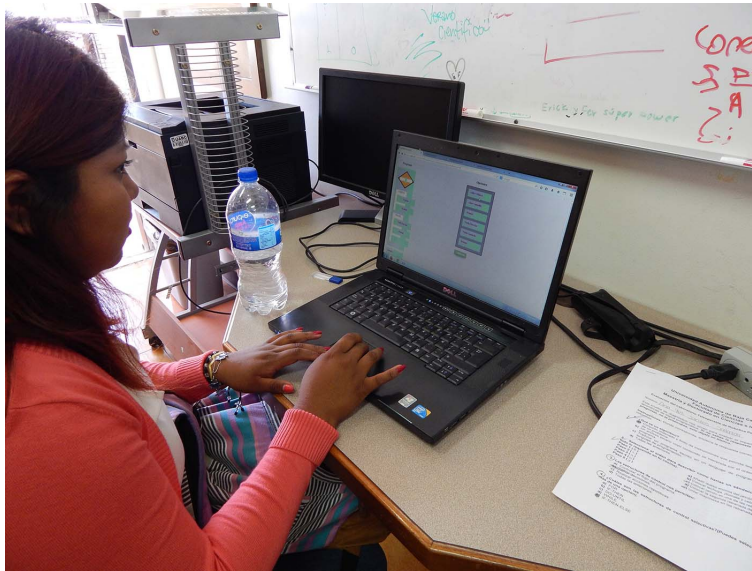


Figura 5.2: Ambiente de programación I.

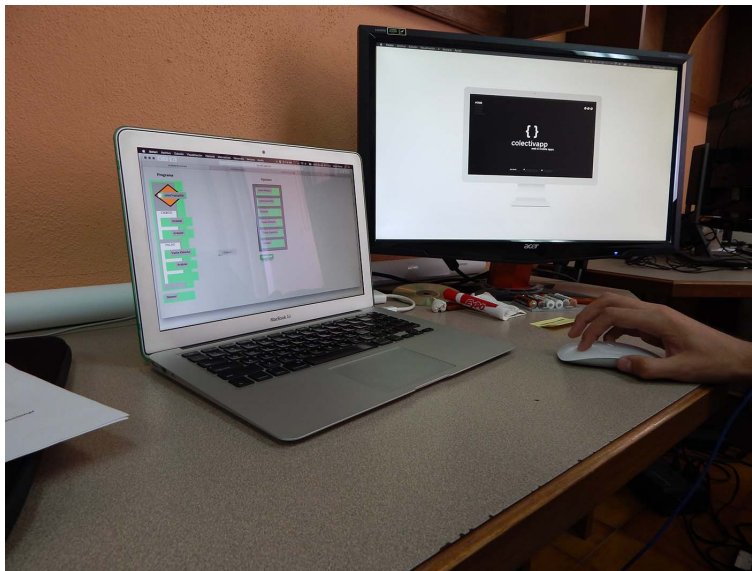


Figura 5.3: Ambiente de programación II.

### 5.5.3.2. Visualización del robot

Con la transmisión en vivo que otorga el LTRP se pudo observar que los participantes realizaban gestos de asombro, al ver que el robot estaba haciendo lo que ellos habían programado, esto generó entusiasmo y motivación para terminar la siguiente práctica más rápido que la anterior y poder observar la ejecución de su código. En la figura 5.4 se muestra la transmisión del vídeo que el alumno ve cuando ejecuta el código.

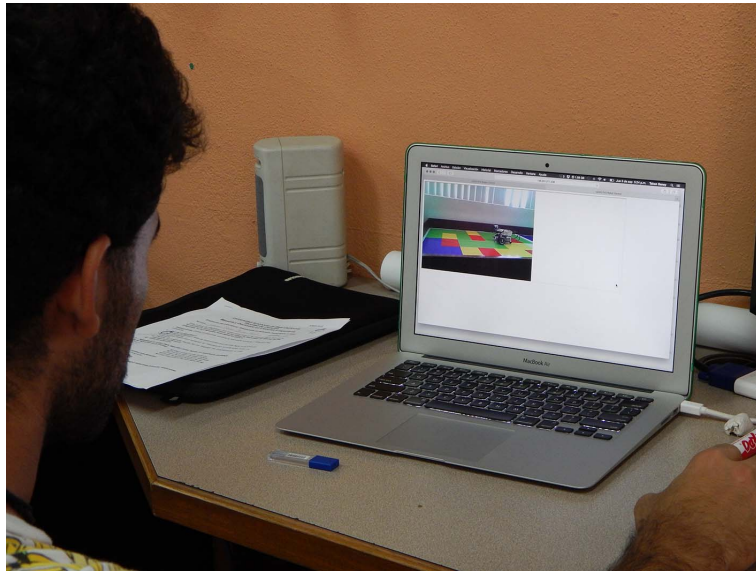


Figura 5.4: Visualización del robot.

# Capítulo 6

## Conclusiones generales

En este capítulo se cierra la investigación del trabajo de tesis, primero describiendo las conclusiones de la tesis, continuando con las aportaciones del trabajo de investigación durante estos dos años, tanto en el ámbito pedagógico como en el tecnológico, principalmente en el área de ciencias computacionales. Por último, se describe el trabajo a futuro de la investigación, orientado a implementar mejoras al LTRP y al ambiente de programación que se desarrolló.

### 6.1. Conclusiones

El uso de las TIC en la educación, en sus diferentes modalidades y aplicándose de manera adecuada, eleva la calidad del proceso de enseñanza-aprendizaje. Las TIC permiten superar barreras del espacio y el tiempo entre docentes y alumnos, ya que los alumnos pueden dirigir su propio aprendizaje fuera del salón y de los horarios establecidos en clase. En este modelo, el proceso de comunicación e interacción permite que exista una instrucción distribuida con base en diferentes fuentes de información, donde se participa de manera activa en el desarrollo del conocimiento y hacia el logro de las metas educativas planteadas [44].

Particularmente, mediante la integración de inteligencia artificial en ambientes de enseñanza basados en tecnologías de información, se han logrado crear sistemas con un alto grado de autonomía, lo que permite tomar decisiones al momento de interactuar con el alumno y consigue personalizar contenidos, estilos y formas de proporcionar instrucción. Asimismo, el uso de los laboratorios remotos y virtuales abre nuevas posibilidades en la docencia al incrementar las oportunidades de experimentación. Así, por ejemplo, en el caso de los laboratorios remotos se extienden los horarios que los equipos están disponibles, realizando un mejor aprovechamiento de los dispositivos, o, en el caso de los laboratorios virtuales, los alumnos interactúan libremente en cualquier momento con modelos simulados de los experimentos. Además de ampliar los horarios de aprendizaje, se permite que los experimentos puedan realizarse sin tener que asistir

físicamente a los laboratorios [45].

Por tanto, los laboratorios remotos y virtuales se constituyen como un nuevo recurso didáctico que puede ser integrado junto a otros componentes pedagógicos dentro de plataformas de eLearning.

Particularmente, con base en las preguntas de investigación que dirigió nuestro estudio 1) ¿Cuáles son las características que deben ser consideradas al elaborar las prácticas interactivas que mantengan el interés del alumno y refuercen el aprendizaje de los temas?; 2) ¿Cuáles son las temáticas de programación que pueden ser implementadas a través de prácticas interactivas en el LTRP?; 3) ¿Qué elementos de las metodologías de diseño instruccional deben ser considerados para la elaboración de las prácticas interactivas en el LTRP?, se logró un resultado aceptable de acuerdo con nuestras expectativas y se llegaron a las siguientes conclusiones. Para la primera pregunta, encontramos que las características que se deben de considerar para realizar las prácticas, deben de ser retadoras para el alumno sin llegar a frustrarlo cuando son muy complicadas o aburrirlo si son muy fáciles, por lo que el nivel de dificultad debe ir incrementado poco a poco. Por parte de la segunda pregunta de investigación, no se abordó en muchas temáticas para aplicarse en el LTRP, solamente se aplicó el tema de estructuras de control selectivas y la construcción de condicionales simples y multiples, pero se dejó la posibilidad de aplicar estructuras de control iterativas. En cuanto a la tercera pregunta, el elemento a considerarse del diseño instruccional para generar prácticas interactivas es la redacción de los objetivos, ya que es importante tener claramente que es lo que se espera que el alumno aprenda al terminar la práctica.

En el presente estudio, obtuvimos que los alumnos que utilizaron el LTRP lograron reforzar su conocimiento en el tema de estructuras de control, aunque se sabe que para poder concluir que el LTRP tienen un impacto significativo en el aprovechamiento educativo, hace falta que se evalúe por más tiempo, con diferentes temas a lo visto en este estudio y con un mayor número de participantes. Otro de los aspectos que lograron los estudiantes al utilizar el LTRP fue la motivación que les generó trabajar con robots y más cuando se puede observar en tiempo real las acciones de los programas que completaron los alumnos, así mismo gracias al ambiente que se desarrolló, los alumnos tomaban las prácticas como retos, ya que se les deba una parte del diagrama y ellos lo tenían que completar.

Por último, cabe señalar que es importante seguir desarrollando materiales, programas o laboratorios utilizando enfoques pedagógicos que ayuden a los maestros a generar mejores prácticas enfocadas a las habilidades y destrezas de los alumnos. Así mismo, pensar en que estas prácticas deben ser didácticas y lúdicas, que permitan que el alumno esté motivado y siga interesado en estudiar una carrera referente a las Ciencias Computacionales.

## 6.2. Aportaciones

- Laboratorio Teleoperable de Robótica Pedagógica, utilizando robots Lego Mindstorm EV3 y cámaras que puedan asignarles una dirección IP.
- Guía para desarrollar material instruccional, esta guía ayuda a los maestros a generar prácticas enfocadas en las habilidades y destrezas de los alumnos.
- Ambiente WEB para generar prácticas para el LTRP, creando un script para traducir código de HTML a XML y de esta forma guardar las prácticas en el servidor WEB.
- Ambiente de programación para completar los diagramas de flujo, creando un script que traduce de XML a HTML para poder mostrarle al alumno la práctica que se diseñó por el maestro.
- Script que traduce de HTML a JAVA para que el robot pueda realizar las acciones que se encuentran en el diagrama de flujo.

## 6.3. Trabajo a futuro

La siguiente lista presenta algunas consideraciones que pueden agregarse a esta tesis, como posibles trabajos futuros:

- Utilizar el LTRP, como herramienta de apoyo para el maestro por más tiempo, con más participantes y con otros temas que se identifiquen con mayor deficiencia por parte de los alumnos, para poder obtener un resultado significativo en el impacto educativo de los alumnos.
- Implementar el LTRP utilizando otros robots educativos, como por ejemplo, brazo robótico, humanoides, otro tipo de configuración al armar el robot de Lego Mindstorm EV3, así como agregarle otros tipos de sensores.
- Realizar la configuración necesaria para poder utilizar el LTRP en una red inalámbrica externa a la de red de la Facultad de Ciencias.

# Apéndice A

## Encuesta

### A.1. Encuesta factores principales que inhiben el aprendizaje de programación

En este apéndice se observa la encuesta utilizada para obtener los principales factores que inhiben el aprendizaje de programación, así como aquellas estrategias de aprendizaje que más le gusta al alumno.

**UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA  
FACULTAD DE CIENCIAS**

**ENCUESTA PARA DETERMINAR FACTORES PRINCIPALES QUE INHIBEN EL  
APRENDIZAJE DE PROGRAMACIÓN**

**Instrucciones:** Favor de marcar con una "X" la opción correspondiente o proporcionar la información solicitada.

1. Proporciona el nombre del municipio, estado, y de la preparatoria de que provienes.

---

---

2. Nombre de la licenciatura que cursas actualmente: \_\_\_\_\_

3. ¿Programaste computadoras antes de entrar a tu carrera actual ?

\_\_\_ SI \_\_\_ NO

*Si la respuesta a esta pregunta fue SI, conteste lo siguiente, de lo contrario pase a la pregunta 6.*

4. ¿Cuánto tiempo llevas programando, en años? \_\_\_\_\_

5. ¿Que lenguajes de programación consideras que dominas?

- |                     |                |
|---------------------|----------------|
| ● Java              | ● Python       |
| ● C/C++             | ● HTML         |
| ● C#                | ● Objective C  |
| ● PHP               | ● MATLAB       |
| ● Javascript/JQuery | ● Otros: _____ |

6. ¿Cómo consideras tu habilidad actual para programar computadoras?

\_\_\_ Ninguna \_\_\_ Baja \_\_\_ Moderado \_\_\_ Avanzado \_\_\_ Experto

Figura A.1: Encuesta factores que inhiben el aprendizaje de programación I.

7. Para cada una de las siguientes situaciones, indica con una X, la complejidad para entenderlas.  
 (MF = Muy Fácil, F = Fácil, M = Moderado, D = Difícil, MD = Muy Difícil).

Situaciones	MF	F	M	D	MD
Diseñar un algoritmo para resolver una tarea.					
Dividir el algoritmo en procedimientos, funciones y/o métodos.					
Encontrar errores en tu propio programa.					
Aprender la sintaxis de un lenguaje.					
Entender estructuras de programación (ciclos, condiciones).					
Utilizar ambientes de desarrollo (Netbeans, Eclipse, Bluejay, etc).					

8. Para cada uno de los siguientes conceptos o estructuras de programación, indica con una X, la complejidad para aprenderlos.  
 (MF = Muy Fácil, F = Fácil, M = Moderado, D = Difícil, MD = Muy Difícil).

Conceptos o Estructuras	MF	F	M	D	MD
Variables					
Arreglos					
Estructuras if-else					
Estructuras de ciclos					
Utilizar librerías					
Manejo de errores					
Abstracción de tipos de datos					
Apuntadores					
Recursividad					

Figura A.2: Encuesta factores que inhiben el aprendizaje de programación II.

9. Para cada una de las siguientes estrategias de aprendizaje , indica con una X, su utilidad en el aprendizaje de programación.  
 (**MU** = *Muy Útil*, **U** = *Útil*, **M** = *Moderado*, **NU** = *No Útil*, **NMU** = *No Muy Útil*).

<b>Estrategias de aprendizaje</b>	<b>MU</b>	<b>U</b>	<b>M</b>	<b>NU</b>	<b>NMU</b>
Lecturas					
Sesiones de práctica en pequeños grupos					
Estudiar por mi cuenta con material didáctico					
Trabajar por mi cuenta en trabajos del curso					
Preguntar al maestro					
Preguntar o consultar en foros/sitios de internet.					
Ver vídeos o tutoriales de cursos (youtube, code.org, codeacademy, etc. )					
Usar cursos en línea masivos y abiertos, como Udacity, Coursera, Edx, Udemy, etc					

10. Para cada uno de los siguientes materiales de apoyo, indica con una X, su utilidad en el aprendizaje de programación.  
 (**MU** = *Muy Útil*, **U** = *Útil*, **M** = *Moderado*, **NU** = *No Útil*, **NMU** = *No Muy Útil*).

<b>Materiales de apoyo</b>	<b>MU</b>	<b>U</b>	<b>M</b>	<b>NU</b>	<b>NMU</b>
Programas de ejemplo					
Notas de lectura					
Ejercicios					
Libro de programación					
Visualización interactiva					

**Preguntas adicionales**

11. ¿Por qué seleccionaste una carrera referente a la computación?

Figura A.3: Encuesta factores que inhiben el aprendizaje de programación III.

12. ¿Qué te motiva aprender a programar?
13. Algunas vez has programado un robot?  SI  NO
14. ¿Si tuvieras la oportunidad de tener un robot, te gustaría diseñar e implementar programas para que resuelva tareas?  SI  NO
15. ¿Crees que utilizar robots en tus prácticas te facilitará el aprendizaje para programar?  SI  NO
16. ¿Consideras que el poder ver las acciones de un robot que programaste, te motivaría a continuar aprendiendo a programar?  SI  NO
17. ¿Te gustaría continuar en la carrera que seleccionaste?  SI  NO
18. Si tienes algún comentario y/o sugerencia, favor de compartirla:

---

---

---

**Nota: Esta es una encuesta anónima, toda la información proporcionada se manejará de forma estrictamente confidencial y con fines académicos.**

Figura A.4: Encuesta factores que inhiben el aprendizaje de programación IV.

# Apéndice B

## Guía para diseñar cursos

### B.1. Introducción

A continuación se presenta una guía para el diseño y planificación de cursos de lógica programática basado en el modelo instruccional de Dick y Carey. Esta guía tiene como propósito acompañar al facilitador paso por paso en la elaboración de material instruccional, a través de ejercicios y actividades, que ayudarán a reforzar el aprendizaje de sus estudiantes. Esta guía en particular esta dirigida a maestros que imparten materias introductorias a la programación, para que sus estudiantes comprendan y apliquen los conceptos básicos de la programación como por ejemplo, la estructuras de control, recursividad, etc. La guía esta organizada de acuerdo a las etapas del modelo instruccional Dick y Carey:

- Identificar metas.
- Análisis instruccional.
- Objetivos.
- Instrumentos de evaluación.
- Estrategia instruccional.
- Materiales instruccionales.
- Evaluación formativa.
- Revisión de instrucción.
- Evaluación sumativa.

## B.2. Identificar metas instruccionales

En este primer paso se identifica qué es lo que se quiere que el estudiante sea capaz de hacer cuando se haya completado la instrucción. El maestro debe de generar una lista priorizada de las metas dependiendo de las dificultades que tengan los estudiantes en la materia. Para ello se tiene que describir la problemática que enfrentan los estudiantes, el propósito de realizar la instrucción y el grupo de estudio a quien va a estar dirigido. La tabla B.1 muestra un ejemplo para identificar las metas para la problemática de aplicar estructuras de control.

Tabla B.1: Ejemplo de la etapa de identificación de metas.

<b>Planteamiento del problema</b>
A los estudiantes de cursos introductorios a la programación se les dificultan algunos conceptos básicos de la programación, en los que se encuentran la construcción y utilización de estructuras de control en sus programas.
<b>Propósito</b>
Definir material instruccional que facilite al estudiante lograr a comprender la construcción y utilización de estructuras de control. Por lo que se proponen las siguiente metas:
<b>Metas</b>
<ul style="list-style-type: none"> <li>▪ Que se familiaricen con conceptos como estructuras de control y adquieran la habilidad para construir estas estructuras.</li> <li>▪ Que los estudiantes adquieran habilidades para aplicar estructuras de control</li> </ul>
<b>Grupo de estudio</b>
El grupo objetivo principalmente son estudiantes que llevan el curso de Introducción a la programación de la carrera de la Licenciatura en Ciencias Computacionales de la UABC.

## B.3. Análisis instruccional

En esta fase se debe identificar aquellas destrezas que deberán enseñarse para lograr la meta o metas instruccionales. En este proceso se obtiene un esquema en la forma de un flujograma que presenta las destrezas y las relaciones entre ellas. La figura B.1 muestra el flujograma del proceso para las metas descritas en el ejemplo de la tabla B.1. Una vez que se ha definido el proceso, este se desglosa en actividades, tal y como se muestra en la figura B.2.

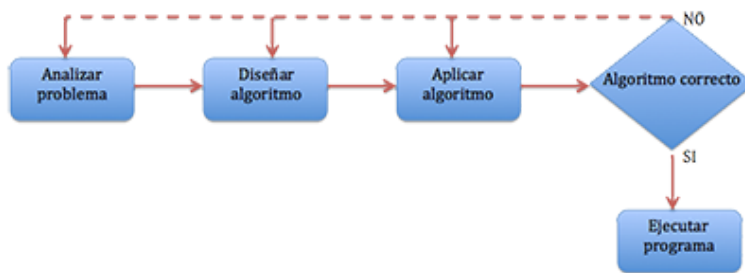


Figura B.1: Proceso para conseguir meta.

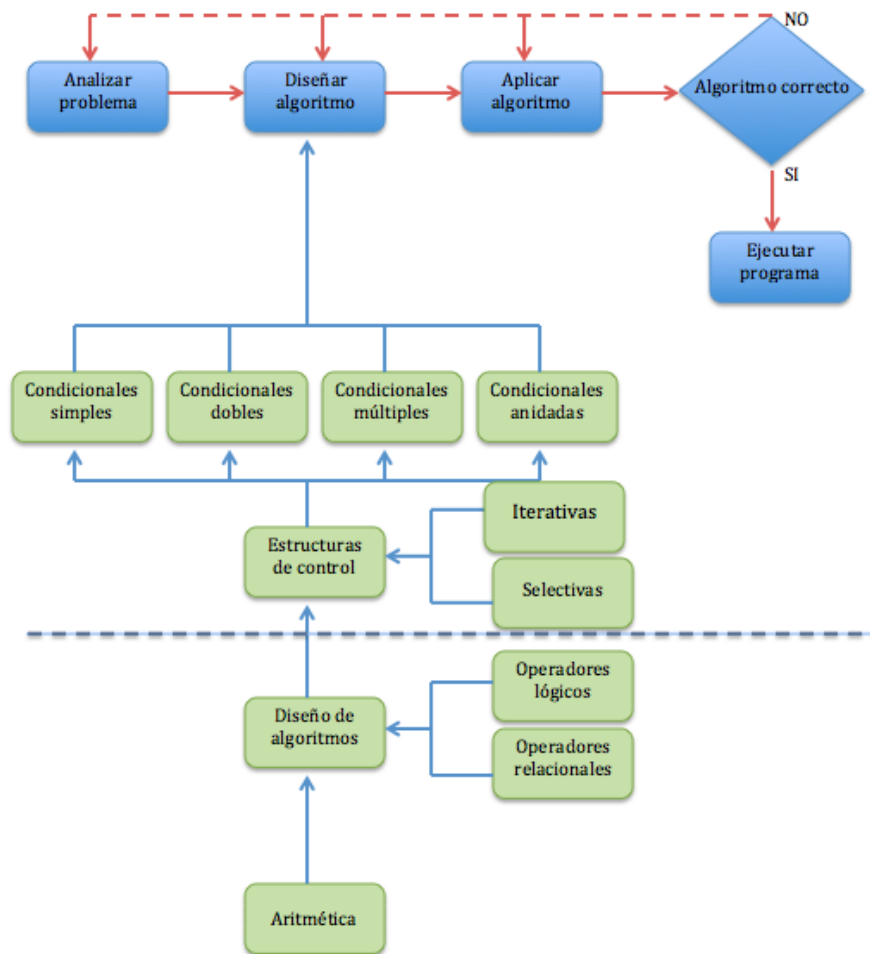


Figura B.2: Proceso para conseguir meta dividida en actividades.

## **B.4. Objetivos y redacción de instrumentos de evaluación**

La tercera y cuarta fase se pueden unir elaborando una tabla con el fin de describir todas las actividades que se deben realizar para cumplir con la meta o las metas propuestas. Se debe describir el objetivo de cada actividad y el método que se va realizar para cumplir dicho objetivo, describiendo la estrategia y/o los instrumentos de evaluación que se utilizarán. La tabla B.2 muestra un ejemplo de como realizar esta tabla, utilizando las actividades que se desglosaron en la figura B.2.

Tabla B.2: Objetivos e instrumentos de evaluación por actividad.

Actividad	Objetivo	Como cumplir el objetivo
2.1 Aritmética	Valorar los conocimientos básicos ya adquiridos de suma, resta, división, multiplicación, igualdad, mayor, menor, etc.	Realizar una valoración de conocimientos.
2.2 Diseño de algoritmos	Valorar los conocimientos básicos ya adquiridos sobre como realizar un buen diseño de algoritmo.	Realizar una valoración de conocimientos.
2.2.1 Operadores relacionales	Distinguir cuales son los operadores relacionales que se pueden utilizar, y como se deben de utilizar.	Realizar una prueba de conocimientos.
2.2.2 Operadores lógicos	Distinguir cuales son los operadores lógicos que se pueden utilizar, y como se deben de utilizar.	Realizar prueba de conocimientos.
2.3 Estructuras de control	Construir correctamente estructuras de control.	Clase teórica de los tipos de estructuras de control que existen, la diferencia que hay entre ellos y como construir la condición.
2.3.1 Selectivas	Saber utilizar los operadores relacionales y lógicos para construir correctamente la condición dentro de una estructura selectiva (if-then, if-then-else).	Clase teórica sobre como utilizar los operadores para construir correctamente la condición dentro de una estructura selectiva.
2.3.2 Iterativas	Saber utilizar los operadores relacionales y lógicos para construir correctamente la condición dentro de una estructura iterativa (while).	Clase teórica sobre como utilizar los operadores para construir correctamente la condición dentro de una estructura iterativa.
2.4 Condicionales simples	Distinguir cuales son las condicionales simples y realizar su construcción.	Realizar prácticas enfocadas en la construcción de una condicional simple.
2.5 Condicionales dobles	Distinguir cuales son las condicionales dobles y realizar su construcción.	Realizar prácticas enfocadas en la construcción de una condicional doble.
2.6 Condicionales múltiples	Distinguir cuales son las condicionales múltiples y realizar su construcción.	Realizar prácticas enfocadas en la construcción de una condicional múltiple.
2.7 Condicionales anidadas	Distinguir cuales son las condicionales anidadas y realizar su construcción	Realizar prácticas enfocadas en la construcción de una condicional anidada.

## B.5. Estrategia instruccional

En esta fase es donde se identifica la estrategia que se utilizará para llevar a cabo la instrucción y se determina cuáles son los medios a utilizarse. En este punto el facilitador debe de realizar una descripción de como llevar a cabo el pre-test y el post-test de las actividades. También debe elaborar una tabla donde se dividen las actividades que se identificaron en el punto B.3, figura B.2 (División de proceso en actividades), por sesión

y por tareas, estimando el tiempo de duración de cada tarea. La tabla B.3 muestra un ejemplo de la descripción del pre-test y pos-test de la problemática de estructuras de control. Y la tabla B.4 muestra el ejemplo de la división de las actividades por sesiones.

Tabla B.3: Diseñar evaluaciones PRE y POST.

<b>Pre-test</b>
<ul style="list-style-type: none"> <li>■ Examen de diagnóstico para identificar el conocimiento sobre Diseño de Algoritmos que tienen los estudiantes (Apéndice C).</li> </ul>
<b>Post-test</b>
<ul style="list-style-type: none"> <li>■ Sesión de prácticas sobre estructuras de control selectivas. Se diseñaron practicas para que el alumno adquiriera conocimientos para construir estructuras de control con una sola condición, estructuras de control con varias condicionales utilizando operadores lógicos y también se diseño una tercera practica para que aprendiera a construir estructuras de control anidadas.</li> <li>■ Examen de aprovechamiento (Apéndice C).</li> </ul>

Tabla B.4: Actividades divididas en sesiones.

Sesión	Actividad	Tiempo
1	Realizar prueba de conocimientos previos	1 hrs.
2	Realizar pre-test de estructuras de control	1 hr.
2	Teoría de estructuras de control	1 hr.
3	Realizar prácticas de estructuras: condicionales simples	2 hr.
4	Realizar prácticas de estructuras: condicionales dobles	2 hr.
5	Realizar prácticas de estructuras: condicionales múltiples	2 hr.
6	Realizar prácticas de estructuras: condicionales anidadas	2 hr.
7	Realizar post-test de estructuras de control	1 hr.

## B.6. Materiales instruccionales

En esta fase el facilitador describe el material que será utilizado en la instrucción, por ejemplo: cuestionarios, presentaciones y exámenes. También se incluyen: vídeos, formatos de multimedia basados en computadoras, y páginas web para la educación a distancia. La tabla B.5 muestra la descripción de los materiales que se utilizaron siguiendo la problemática de estructuras de control.

Tabla B.5: Material instruccional.

Material	Descripción
Examen de diagnóstico de Diseño de algoritmos	Examen para saber el conocimiento que tienen los alumnos en diseñar algoritmos.
Prácticas estructuras de control selectivas	Prácticas enfocadas a que el alumno aprenda a construir estructuras de control selectivas con condicionales simples, con varias condiciones utilizando operadores lógicos y aprender a construir estructuras de control anidadas.

## B.7. Evaluación formativa

Una vez que se finalice con la elaboración de la instrucción, se deberán recoger los datos (resultados del pre-test, resultados del post-test, resultados de las prácticas que se realizaron, etc.) para así mejorarla. El facilitador lleva a cabo tres evaluaciones: evaluación uno-a-uno, evaluación de grupo pequeño y evaluación de campo. Cada tipo de evaluación le provee al profesor información valiosa para mejorar la instrucción. La evaluación formativa también puede ser aplicada a los materiales instruccionales y a la instrucción en el salón de clases. Es importante saber que la evaluación formativa puede darse en cualquier o en todas las fases del modelo. La tabla B.6, presenta el formato de la revisión de materiales utilizados en la instrucción.

Tabla B.6: Revisión de materiales.

Componente	Problema	Cambio	Justificación
Lista de los materiales utilizados en la instrucción, como por ejemplo, presentaciones, prácticas, herramienta tecnológica, etc.	Se describe brevemente el problema de cada material que el estudiante haya percibido y lo ha dado a conocer.	Una descripción breve del cambio que se podría hacer al material dependiendo de la problemática identificada.	Se justifica el cambio que se va a realizar añadiendo algunas observaciones de los estudiantes, o sino hubo cambio al material se describe el porqué no se realizó un cambio.

## B.8. Revisión de instrucción

Esta fase es similar a la del punto B.7 (Evaluación formativa), la diferencia es que en esta fase, se hace un resumen y un análisis a base de los datos recogidos en la fase de la evaluación formativa. Se re-examina la validez del análisis instruccional, las conductas

de entrada, los objetivos, etc., y finalmente, se incorporan las revisiones para hacer la instrucción más efectiva.

## **B.9. Conclusión**

Utilizando diseño instruccional nos ayuda asegurar que los estudiantes alcancen los objetivos planteados por el facilitador, diseñando prácticas enfocadas en las necesidades y las metas a cumplir, basándonos en el modelo instruccional Dick y Carey.



# Apéndice C

## Pre-test y Post-test

JUNIO 2015

**Universidad Autónoma de Baja California  
Facultad de Ciencias  
Maestría y Doctorado en Ciencias e Ingeniería**

**Evaluación del Laboratorio Teleoperable de Robótica Pedagógica (LTRP)**

Nombre: \_\_\_\_\_ Semestre: \_\_\_\_\_

Esta prueba de diagnóstico nos sirven para evaluar el impacto del LTRP en tu aprendizaje de lógica programática. *Si no sabes la respuesta de alguna de las preguntas, no te preocupes, solo déjala en blanco.*

**Instrucciones:** Donde corresponda, encierra la respuesta correcta o relaciona las columnas.

**1.- ¿Qué es un algoritmo?**

- a) Conjunto ordenado y finito de instrucciones que permite llegar a la solución de un problema.
- b) Un archivo binario cuyo contenido se interpreta por el ordenador como un programa.
- c) Traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación.
- d) Todas las anteriores.

**2.- Selecciona el orden para describir como harías un sándwich (teniendo todo lo requerido en tu cocina).**

- |                   |   |
|-------------------|---|
| <b>Paso 1</b> ( ) | a) Añadir queso encima del jamón          |
| <b>Paso 2</b> ( ) | b) Colocar lechuga, jitomate y cebolla    |
| <b>Paso 3</b> ( ) | c) Colocar jamón en una de las rebanadas  |
| <b>Paso 4</b> ( ) | d) Colocar la segunda rebanada de pan     |
| <b>Paso 5</b> ( ) | e) Untar mayonesa en las rebanadas de pan |

**3.- Las estructuras de control nos permiten:**

- a) Controlar el flujo del programa.
- b) Tomar decisiones.
- c) Realizar acciones repetitivas.
- d) Todas las anteriores.

**4.- ¿Cuáles son las estructuras de control selectivas?(Puedes seleccionar más de una opción).**

- a) FOR
- b) IF-THEN
- c) DO-UNTIL
- d) WHILE
- e) IF-THEN-ELSE

Figura C.1: Examen pre-test y post-test I.

JUNIO 2015

5.- Es martes y deseas ir al cine, realiza el pseudocódigo del proceso requerido para comprar tu boleto. Debes de utilizar al menos una estructura de control IF-THEN ó IF-THEN-ELSE. (No es necesario utilizar todas las opciones).

- |             |  |
|-------------|--|
| Paso 1 ( )  | a) ver película                        |
| Paso 2 ( )  | b) seleccionar horario                 |
| Paso 3 ( )  | c) IF(película está en cartelera) THEN |
| Paso 4 ( )  | d) ELSE seleccionar otro horario       |
| Paso 5 ( )  | e) seleccionar película                |
| Paso 6 ( )  | f) IF(dinero >= costo boleto) THEN     |
| Paso 7 ( )  | g) ELSE seleccionar otra película      |
| Paso 8 ( )  | h) pagar boleto                        |
| Paso 9 ( )  | i) llegar al cine                      |
| Paso 10 ( ) | j) entrar a la sala                    |

6.- Elige la condición escrita correctamente

- a) IF(10)
- b) WHILE(7\*3)
- c) IF(color i= "rojo")
- d) IF(color == "azul")

7.- Elige la condición escrita correctamente

- a) IF(distancia == 50 && color != "amarillo")
- b) WHILE(distancia > 100 & < 100)
- c) WHILE(color = "verde" || color i "rojo")
- d) IF(distancia == 50 & color != "amarillo")

8.- Realiza un pseudocódigo donde incluyas una condición múltiple (Escenario: comprar boleto del cine. No es necesario utilizar todas las opciones).

- |             |   |
|-------------|---|
| Paso 1 ( )  | a) IF(película en cartelera && hora <= horario película) THEN |
| Paso 2 ( )  | b) ELSE seleccionar otro horario                              |
| Paso 3 ( )  | c) seleccionar película                                       |
| Paso 4 ( )  | d) ver películas en cartelera                                 |
| Paso 5 ( )  | e) IF(dinero >= costo boleto) THEN                            |
| Paso 6 ( )  | f) IF(dinero <= costo boleto) THEN                            |
| Paso 7 ( )  | g) pagar boleto con efectivo                                  |
| Paso 8 ( )  | h) IF(tengo efectivo && dinero >= costo boleto) THEN          |
| Paso 9 ( )  | i) entrar a la sala   |
| Paso 10 ( ) | j) llegar al cine   |
|             | k) ver película   |
|             | l) ELSE pagar boleto con tarjeta                              |

Figura C.2: Examen pre-test y post-test II.

JUNIO 2015

9.- ¿Cuál de las siguientes opciones es una estructura de control anidada?

- |  |   |
|--|---|
| <p>a) IF(condición 1){<br/>    CÓDIGO 1<br/>}<br/>IF(condición 2){<br/>    CÓDIGO 2<br/>}</p>  | <p>b) IF(condición 1 &amp;&amp; condición 2){<br/>    CÓDIGO<br/>}</p>  |
| <p>c) IF(condición 1    condición 2){<br/>    IF(condición 3){<br/>        CÓDIGO 1<br/>    }<br/>    ELSE{<br/>        CÓDIGO 2<br/>    }<br/>}</p> | <p>d) IF(condición 1){<br/>    CÓDIGO 1<br/>}<br/>IF(condición 2 &amp;&amp; condición 3){<br/>    CÓDIGO 2<br/>}<br/>ELSE{<br/>    CÓDIGO 3<br/>}</p> |

10.- Realiza un pseudocódigo donde incluyas una estructura de control anidada (Escenario: comprar boleto del cine. No es necesario utilizar todas las opciones).

- |             |   |
|-------------|---|
| Paso 1 ( )  | a) IF(película en cartelera && hora <= horario película) THEN |
| Paso 2 ( )  | b) IF(tengo efectivo && dinero >= costo boleto) THEN          |
| Paso 3 ( )  | c) IF(hora <= horario película) THEN                          |
| Paso 4 ( )  | d) IF(dinero >= costo boleto) THEN                            |
| Paso 5 ( )  | e) IF(hora >= horario película) THEN                          |
| Paso 6 ( )  | f) IF(película en cartelera) THEN                             |
| Paso 7 ( )  | g) ELSE seleccionar otro hora                                 |
| Paso 8 ( )  | h) llegar al cine   |
| Paso 9 ( )  | i) ver película   |
| Paso 10 ( ) | j) seleccionar película                                       |
|             | k) seleccionar horario  |
|             | l) ver películas en cartelera                                 |
|             | m) pagar boleto con efectivo                                  |
|             | n) ELSE pagar con tarjeta                                     |
|             | ñ) IF(tengo efectivo) THEN                                    |
|             | o) ELSE seleccionar otra película                             |
|             | p) pagar boleto con tarjeta                                   |
|             | q) ELSE IF(tengo tarjeta) THEN                                |
|             | r) entrar a la sala   |
|             | s) IF(tengo tarjeta) THEN                                     |

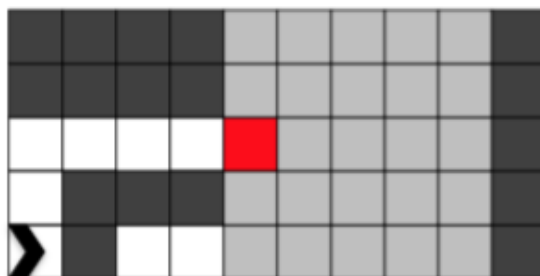
Figura C.3: Examen pre-test y post-test III.

JUNIO 2015

**11.- Resuelve el siguiente ejercicio.**

En Marte, al *robot explorador* se le han descompuesto las celdas solares para recargar su batería, por lo que tiene que regresar a su base para reemplazarlas. Tu objetivo es programar los pasos necesarios para que el robot conduzca por el camino blanco y llegue a tiempo a su base (celda roja). La posición donde está localizado el robot y su orientación se indica con la flecha negra. Nótese que cada celda es equivalente a un "avanzar".

**Nota:** Considera que este robot no cuenta con la función de dar vuelta a la izquierda, por lo que tienes que usar las opciones proporcionadas.



PROGRAMA	OPCIONES
<input type="text"/>	AVANZAR
AVANZAR	DETENER
<input type="text"/>	VUELTA
<input type="text"/>	VUELTA
<input type="text"/>	VUELTA
<input type="text"/>	VUELTA
<input type="text"/>	
<input type="text"/>	

Figura C.4: Examen pre-test y post-test IV.

# Bibliografía

- [1] Gabriel Hernández López. El origen de la Robótica Industrial y su desarrollo. *Revista Oficial de ISA México Sección Central*, pages 18–22, 2008.
- [2] Adolfo Obaya Valdivia. El construccionismo y sus repercusiones en el aprendizaje asistido por computadora. *ContactoS*, 48:61–64, 2003.
- [3] Manuel Villarruel-Fuentes. El constructivismo y su papel en la innovación educativa. *Revista de Educación y Desarrollo*, 20, 2012.
- [4] Enrique Ruiz-Velasco Sánchez. Robótica pedagógica virtual para la inteligencia colectiva, 2010.
- [5] A Barrón Ruiz. Aprendizaje por descubrimiento: principios y aplicaciones inadecuadas. *Enseñanza de las Ciencias*, 11(1):3–11, 1993.
- [6] Enrique Ruiz-Velasco Sánchez. *EDUCATRÓNICA. Innovación en el aprendizaje de las ciencias y la tecnología*. Madrid, 2007.
- [7] Alejandra Velasco Pérez, J Jesús Arellano Pimentel, José Vicente Martínez, and Salma L. Velasco Pérez. Laboratorios virtuales: alternativa en la educación, 2013.
- [8] Francisco A. Candelas, Fernando Torres, Pablo Gil, Francisco Ortiz, Santiago Puente, and Jorge Pomares. Laboratorio virtual remoto para robótica y evaluación de su impacto en la docencia, 2004.
- [9] Rogelio Prieto-Alvarado, Roberto Bernal-Guadiana, and Ulises Zaldívar-Colado. Creación de un Laboratorio Virtual para Optimizar el uso de un Laboratorio de Robótica Real, 2010.
- [10] Luis M Jiménez, Rafael Puerto, Óscar Reinoso, César Fernández, and Ramón Ñeco. RECOLAB: Laboratorio remoto de control utilizando matlab y simulink. *Revista Iberoamericana de Automática e Informática Industrial*, 2(2):64–72, 2005.
- [11] Luis Payá, Oscar Reinoso, Arturo Gil, and Luis M. Jiménez. Plataforma Distribuida para la Realización de Prácticas de Robótica Móvil a través de Internet. *Información Tecnológica*, 18(6):27–38, 2007.

- [12] Dictino Chaos, Jesús Chacón, Jose Antonio Lopez-Orozco, and Sebastián Dormido. Virtual and remote robotic laboratory using EJS, MATLAB and LabVIEW. *Sensors*, 13:2595–2612, January 2013.
- [13] BAN Ki-moon. *Information economy report 2011. ICTs as an Enabler for Private Sector Development*. Unctad, 2011.
- [14] BAN Ki-moon. *Information economy report 2012. The Software Industry and Developing Countries*. Unctad, 2012.
- [15] Ásrún Matthíasdóttir. Learning objects in a multimedia interactive environment . The Codewitz project, 2004.
- [16] Jens Bennedsen and Michael E. Caspersen. Failure rates in introductory programming. *ACM SIGCSE Bulletin*, 39(2):32–36, June 2007.
- [17] Asociación Nacional de Universidades e Instituciones de Educación Superior (México). *Anuario estadístico 2004: Población escolar de licenciatura y técnico superior en universidades e institutos tecnológicos*. Asociación Nacional de Universidades e Instituciones de Educación Superior, 2006.
- [18] Ásrún Matthíasdóttir. How to teach programming languages to novice students ? Lecturing or not ?, 2006.
- [19] Ásrún Matthíasdóttir. What students find difficult in learning programming, 2004.
- [20] Essi Lahtinen, Kirsti Ala-Mutka, and Hannu-Matti Järvinen. A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, 37(3):14–18, September 2005.
- [21] Walter Dick, Lou Carey, and James O Carey. The Systematic Design of Instruction. In *The Systematic Design of Instruction*, chapter 1, pages 2–12. Pearson Education, fifth edition, 2001.
- [22] Peter Williams, Lynne Schrum, Albert Sangrà, and Lourdes Guàrdia. Fundamentos del diseño técnico-pedagógico en e-learning, 2009.
- [23] Consuelo Belloch. Diseño instruccional. 11, 2013.
- [24] Frida Díaz-Barriga Arceo and Gerardo Hernández Rojas. Estrategias docentes para un aprendizaje significativo. una interpretación constructivista. 2<sup>a</sup>. ed.) México: McGraw Hill, 2002.
- [25] Ronald Feo. Orientaciones básicas para el diseño de estrategias didácticas. *Tendencias pedagógicas*, 16(1):220–236, 2010.

- [26] Patricia Witham K, Olga Mora M, and M Teresa Sánchez. Estilos de Aprendizaje en Estudiantes Universitarios de Concepción. *Revista Electrónica de Trabajo Social*, 7, 2008.
- [27] Francisco Camarero Suárez, Francisco Martín del Buey, and Javier Herrero Diez. Estilos y estrategias de aprendizaje en estudiantes universitarios. *Psicothema*, 12(4):615–622, 2000.
- [28] Cristiane Camilo Hernandez, Luciano Silva, Rafael Alencar Segura, Juliano Schimiguel, Manuel F. Paradela Ledón, Luis N. Mendes, Bezerra, and Ismar Frango Silveira. Teaching Programming Principles through a Game Engine. *CLEI Electronic*, 13(2):1–8, 2010.
- [29] R Burbaite, V Stuikeys, and R Marcinkevicius. The LEGO NXT Robot-based e-Learning Environment to Teach Computer Science Topics. *Elektronika ir Elektrotechnika*, 18(9):113–117, 2012.
- [30] Kim B Bruce. Controversy on How to Teach CS 1 : A Discussion on the SIGCSE-members Mailing List. *ACM SIGCSE Bulletin*, 36(4):29–35, 2004.
- [31] Secretaría de Educación Pública (SEP). *Programa sectorial de la educación*. Comisión Nacional de Libros de Texto Gratuitos, México, 2007.
- [32] Bari Courts and Jan Tucker. Using Technology To Create A Dynamic Classroom Experience. *Journal of college teaching and learning*, 9(2):121–128, 2012.
- [33] Anabela Gomes and António J. Mendes. A study on student performance in first year CS courses, 2010.
- [34] Adrian Enciso, Jose A. González-Fraga, and Omar Álvarez Xochihua. Karelego.
- [35] Nelson D. García Hurtado, Leonardo F. Castillo García, and Anthony J. Escobar Jiménez. Plataforma robótica educativa ROBI. *Revista Colombiana de Tecnologías de Avanzada*, 1(19):140–144, 2012.
- [36] David T Butterworth. Teaching C / C ++ Programming with Lego Mindstorms, 2012.
- [37] Isidro Calvo and Gorka Perianez. Uso conjunto de la plataforma Mindstorm NXT y metodologías PBL en Informática Industrial, 2010.
- [38] Aldebaran. Nao H25.
- [39] Francisco A. Candelas Herías and José Sánchez Moreno. Recursos didácticos basados en Internet para el apoyo a la enseñanza de materias de área de Ingeniería de Sistemas y Automática. *Revista Iberoamericana de Automática e Informática Industrial*, 2(2):93–101, 2005.

- [40] Barney Dalgarno, Andrea G. Bishp, and Danny R. Bedgood Jr. The potential of virtual laboratories for distance education science teaching : reflections from the development and evaluation of a virtual chemistry laboratory, 2003.
- [41] Mónica González, Fabiana Ferreira, and Guillermo Casas. La experimentación en ambientes virtuales : aplicación de estrategias de aprendizaje para una formación en competencias en Ingeniería en Electrónica , Automatización y Control ., 2011.
- [42] Akar Ayse, Yayla; Aynur. Web based real tieme remote laboratory with Lab-View access for analog and digital comunication courses. *Journal of electrical and electronics engineering*, 8(2):671–681, 2008.
- [43] Roberto Hernández Sampieri, Carlos Fernández Collado, and Pilar Baptista Lucio. Definiciones de los enfoques cuantitativo y cualitativo, sus similitudes y diferencias. In Mc Graw Hill, editor, *Metodología de la investigación*, chapter 1, pages 2–21. México, fifth edition, 2010.
- [44] T.C. Islas and E. Martínez. El uso de las tic como apoyo a las actividades docentes. *Revista RED*, (204):30–35, 2008.
- [45] I. Calvo, M. Marcos, D. Orive, and I. Sarachaga. Building complex remote learning laboratories, 2008.