



INSTITUTO UNIVERSITARIO AERONAUTICO

Proyecto de Grado

Elaborado por:
Nicolás Osvaldo Guini

Facultad de Ingeniería
Carrera Informática

Instituto Universitario Aeronáutico
Córdoba, Julio de 2015

Dedicatorias

A toda mi familia, principalmente a mi Madre y mi Padre ,,,

... quienes siempre me respaldaron para poder llegar a esta instancia en mis estudios, fueron ellos los que me alentaron y estuvieron siempre apoyándome moral y psicológicamente.

A mis amigos ...

... porque permanecieron atentos a mi progreso, y me ayudaron desinteresadamente cuando los necesité; me acompañaron a lo largo de este proceso, estuvieron pendientes de mis aciertos y desaciertos. Agradezco la confianza que depositaron en mí y en mi proyecto, por creer que podía y haber hecho de mi etapa universitaria un trayecto de vivencia que nunca voy a olvidar. Que Dios bendiga cada uno de nuestros caminos, nos llene de fortalezas y virtudes y porque no, desear que algún día esos caminos se crucen, si el destino así lo quiere.

Agradecimientos

A mi Tutor de Tesis Ing. Mariano A. Garcia Mattio ...

... por la orientación y ayuda que me brindó para la realización de esta tesis, por su apoyo y amistad que me permitieron aprender mucho más allá de lo que se estudia en los textos.

A mi Tutor inicial, Mgtr. Pablo Piccolotto ...

... con él empecé este trabajo de tesis, y fue quien a partir de una idea que tenía, me alentó e impulsó para presentarlo.

A todos mis Profesores de la Carrera ...

... que me enseñaron tanto de la profesión como de la vida, estimulándome siempre a seguir adelante.

A todas las personas involucradas directa o indirectamente durante el desarrollo de esta tesis.

PROGRAMA INCENTIVO
NIVEL MEDIO
DE PROGRAMACIÓN: ROBOCODE

Índice

Tabla de Contenido

Resumen	8
Objetivos.....	12
Objetivo General.....	12
Objetivos Específicos	12
Desarrollo	13
Fundamentación del Proyecto	13
Los Modelos Pedagógicos.....	13
La Tecnología en las escuelas	16
Educación Tecnológica	16
Tecnología Educativa.....	18
Acercar la Programación a las escuelas	20
Estrategias Metodológicas.....	22
Instrumentos: Elementos de la Programación Básica	26
Aspectos Técnicos de Java y Robocode	29
Java	29
Robocode.....	34
Jornadas de Capacitación de Capacitadores	40
Introducción	40
Objetivos.....	42
Temario	42
Metodología.....	42
Planificación de las Jornadas	43
Contenidos	43
A la hora de evaluar a sus estudiantes	46
Síntesis.....	48
Programa Incentivo Nivel Medio de Programación: Robocode	49
Introducción	49
Objetivos.....	50

Estructura del Taller.....	50
Distribución de horas	51
Contenidos	52
Módulo Java.....	52
Conocimientos mínimos de POO	53
Módulo Robocode	53
Ejercicios.....	55
Metodología.....	56
Evaluación: Condiciones para la aprobación	56
Síntesis.....	57
Antecedentes.....	58
Experiencia piloto Universitaria	58
Identificar nuestro Proyecto con los resultados obtenidos de la Experiencia piloto Universitaria en el Nivel Medio	59
Aspectos que caracterizaron a alumnos de las escuelas que tomaron la capacitación y realizaron la competencia.....	60
Aspectos desarrollados por los alumnos en la capacitación y la competencia	61
Análisis de Resultados	62
Conclusiones.....	64
Referencias / Bibliografías.....	66
Anexos.....	68
Anexo 1 – Programa Incentivo Nivel Medio de Programación: Robocode - Ejercicios	69
Anexo 2 – Códigos de Computación	72
Anexo 3 – Reglamento de las Olimpiadas Robocode.....	75
Anexo 4 - Ejemplo de Cronograma de Actividades	78
Anexo 5 – Certificados Otorgados.....	79
Anexo 6 – Formularios de Inscripción.....	82

Listado de símbolos y convenciones

No Aplica

Resumen

Con este Proyecto se demuestra la viabilidad de insertar el lenguaje de Programación en los colegios secundarios, el cual es una de las herramientas para incentivar a los adolescentes a seguir una carrera universitaria.

Estamos transitando el siglo 21, la era digital, la sociedad de la creatividad, las universidades fueron creando muchísimas carreras en torno a la tecnología formando excelentes profesionales en distintas áreas; nuestros adolescentes también están ávidos de tecnología, es el momento de que las universidades interactúen con el nivel medio.

Aprovechando que la tecnología está inserta en casi todas las actividades diarias del ser humano y, avanzando día a día, insertamos la actividad lúdica en el aula y la propuesta de una competencia entre colegios para atraer tanto a los docentes como a los alumnos a renovarse permanentemente en esta área.

El punto de partida que utilizamos son las competencias Robocode, las cuales se realizan sobre un software gratuito y crea una estrategia innovadora en el aula. Como instancia previa y necesaria para acceder a ésta, se instruirá sobre Programación Java.

Así, docentes y alumnos podrán generar actividades de investigación, diseño, codificación, solución de problemas, etc., utilizando elementos que ya son cotidianos para los adolescentes y lograr a través de su ingenio y creatividad ganar para su colegio las Olimpíadas Robocode.

Para dar fortaleza a este proyecto iniciamos nuestra investigación desde las diferencias entre Educación Tecnológica y Tecnología Educativa, pasando por los modelos pedagógicos y las estrategias metodológicas apropiadas para la inserción del Programa Incentivo de Programación en la currícula secundaria.

Reforzamos nuestro trabajo de campo, analizando una experiencia piloto que realizó el Instituto Universitario Aeronáutico (IUA) con colegios del nivel medio y aprovechando que la mayoría de las escuelas secundarias forman parte del plan nacional "Conectar igualdad" (que provee conexión a Internet y netbooks) proponemos Jornadas de Capacitación para los docentes y un Taller de Programación en los colegios.

A través del marketing educativo llegaremos a los colegios con la invitación a participar en estas competencias, motivando el estudio de lenguajes de programación.

Consideramos que con este proyecto, contribuimos al desarrollo de una formación integral en esta sociedad digital del siglo 21, que requiere de nuevas habilidades y destrezas vinculadas a diseñar, crear e investigar con los nuevos medios digitales.

Palabras Claves

- **Educación Tecnológica:** Se refiere al conocimiento específico y socialmente significativo.
- **Java:** Lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener pocas dependencias de implementación como fuera posible.
- **Marketing Educativo:** Se denomina Marketing Educativo a todas las acciones que ejecuta una institución de educación para promover y desarrollar toda su oferta académica.
- **Objeto:** Son datos y el código (funciones o métodos) que se combinan en entidades llamadas objetos.
- **Robocode:** Robocode es un juego educativo Open Source escrito en Java, en el que los competidores deben diseñar y programar sistemas de control para unos tanques (robots), que compiten con el objetivo de eliminar el resto de los adversarios.
- **Simulador:** Herramienta interactiva que permite capacitar y entrenar a los alumnos en un entorno muy similar al real.
- **Taller:** Un taller es una modalidad de enseñanza y estudio caracterizada por la actividad, la investigación operativa, el descubrimiento científico y el trabajo en equipo que, en su aspecto externo, se distingue por el acopio, la sistematización y el uso de material especializado acorde con el tema para la elaboración de un producto tangible.
- **Tecnología Educativa:** Es el conjunto de tecnologías de gestión y de técnicas de actuación que utilizan las sociedades en determinados momentos históricos para difundir el conocimiento acumulado y socialmente significativo, desde la perspectiva de los grupos responsables de legislar, organizar y conducir.
- **TICs:** Tecnologías de la Información y Comunicación, se refiere a la preparación que tienen estudiantes para satisfacer las necesidades de tecnologías en cómputo y comunicación de gobiernos, seguridad social, escuelas y cualquier tipo de organización.

Introducción

Que todos aprendamos a leer y a escribir no quiere decir que todos nos convertiremos en escritores, que los alumnos de nivel medio aprendan a programar no significa que todos serán programadores.

La influencia de la tecnología en el mundo contemporáneo es tal, que hoy podemos afirmar que no hay aspecto de la vida humana que no esté influido por el desarrollo tecnológico.

La misma industria fue incrementando rápidamente la utilización de herramientas, máquinas y técnicas modernas para aumentar sustancialmente la cantidad y la calidad de sus productos. El desarrollo industrial necesita cada vez más acrecentar su tecnificación y esto trae aparejado la especialización de los operarios y de los técnicos.

Teniendo en cuenta las estadísticas de proyecciones laborales, podemos ver que las industrias tecnológicas son las más atractivas y de mayor crecimiento en el mundo actual. Pues, a lo largo de la historia y según las necesidades que se les presentaban, los seres humanos fueron creando y construyendo un sinnúmero de productos tecnológicos. Con sólo fijar la atención a nuestro alrededor podemos constatar que prácticamente estamos rodeados de objetos tecnológicos como por ejemplo: bancos, sillas, cuadernos, lapiceras, radios, automóviles, televisores, computadoras, celulares, etcétera.

La tecnología condiciona nuestras actividades, nuestro comportamiento, el desarrollo social y, en consecuencia, nuestra cultura. Sin temor a equivocarnos podemos decir que vivimos en un mundo artificial, producto de la creación y construcción humana. La complejidad y la extensión que ha adquirido el mundo artificial plantean el riesgo de aislar e incomunicar al hombre, impidiéndole la percepción del mundo natural.

El hombre, frente a la magnitud del mundo tecnológico que creó, debe ser el dueño y no el esclavo. En consecuencia, para poder tener el control necesita saber: el porqué, el cómo y el para qué de su funcionamiento. Por ende, resulta necesaria e indispensable la inclusión de Educación Tecnológica en la Escuela.

El Sistema Educativo Argentino introduce la Tecnología como contenido de aprendizaje. Dentro de este campo podemos incluir (ver Colección educ.ar):

-  **La educación tecnológica.** Se refiere al conocimiento específico y socialmente significativo.
-  **La tecnología educativa.** Es el conjunto de tecnologías de gestión y de técnicas de actuación que utilizan las sociedades en determinados momentos históricos para difundir el conocimiento acumulado y socialmente significativo, desde la perspectiva de los grupos responsables de legislar, organizar y conducir.

Desde este Proyecto intentamos acercar la Programación a las escuelas secundarias entendiendo que la Programación tiene un rol central y contribuye al desarrollo del “pensamiento computacional”, promueve el pensamiento analítico, sistemático, fomenta la creatividad y el trabajo colaborativo, todas estas habilidades son consideradas fundamentales para la sociedad del siglo 21, llamada la “Sociedad de la Creatividad”. Este aprendizaje es posible trasladarlo/aplicarlo a áreas disciplinares diferentes a las específicas de informática.

Objetivos

Objetivo General

- ⊕ Desarrollar un modelo de referencia de Programa Incentivo para Nivel Medio de Programación, utilizando la plataforma Robocode con lenguaje Java.

Objetivos Específicos

- ⊕ Instruir a los Docentes capacitadores, del área de tecnología del nivel medio, sobre el lenguaje de Programación Java haciendo eje en el Software Robocode como entorno propicio para el desarrollo de propuestas de enseñanza de forma lúdica.
- ⊕ Enseñar a los alumnos de las escuelas secundarias (con o sin orientación técnica) para que puedan:
 - ⊕ Comprender y elaborar ideas sobre Programación.
 - ⊕ Ser capaces de programar robots en java.
 - ⊕ Aplicar la creatividad en pos de crear algoritmos que representen estrategias de combate del robot.

Desarrollo

Fundamentación del Proyecto

Para despertar vocaciones tempranas, la mayoría de las facultades de ingeniería del país han emprendido diversos programas en las escuelas secundarias.

Con el presente Proyecto, buscamos fortalecer los lazos entre la Universidad y el Nivel Medio a través de una **Propuesta de Aprendizaje de Programación de manera lúdica**, en donde el estudiante (en una actividad tecnológica) aplique su ingenio y creatividad en el desarrollo de piezas de software que representen robots que competirán en una olimpiada.

Para llevar a cabo esta idea, necesitamos contar con personal idóneo en el uso de estos procesos tecnológicos, razón por la cual se realizarán “Jornadas de Capacitación de Capacitadores” cuyo propósito es promover y estimular en los educadores la potencialidad que la tecnología posee para las prácticas de Programación de forma lúdica.

También, como dijimos anteriormente, despertar vocaciones tempranas en los alumnos del nivel medio, particularmente en el aprendizaje de la programación y la algoritmia, para ello, elaboraremos un programa en el cual los alumnos deberán definir y programar un robot sobre la plataforma Robocode, en lenguaje Java.

Luego de que cada alumno diseñe, elabore y codifique su propio robot, se enfrentará a su segundo desafío, que es el evento principal: **Las Olimpiadas de Robocode**.

Los Modelos Pedagógicos

Un Modelo Educativo consiste en una recopilación o síntesis de distintas teorías y enfoques pedagógicos, que orientan a los docentes en la elaboración de los programas de estudios y en la sistematización del proceso de enseñanza y aprendizaje. En otras palabras es un patrón conceptual a través del cual se esquematizan las partes y los elementos de un programa de estudio.

Es evidente la diversidad de conceptos determinantes asociados a la definición de modelo pedagógico. Resumiremos en lo siguiente:

- ✓ *Implica el contenido de la enseñanza, el desarrollo del niño y las características de la práctica docente.*
- ✓ *Pretende lograr aprendizajes y se concreta en el aula.*
- ✓ *Instrumento de la investigación de carácter teórico creado para reproducir idealmente el proceso enseñanza - aprendizaje.*
- ✓ *Paradigma que sirve para entender, orientar y dirigir la educación.*

Dentro de los modelos pedagógicos podemos encontrar: Tradicional, Conductista, Cognoscitivista, Socialista, Constructivista, Conceptual y el de Tecnología Educativa NTIC.

Tabla Comparativa de los Modelos Pedagógicos respecto al papel que deben cumplir la educación, el docente y el estudiante en cada uno de ellos

MODELO	EDUCACIÓN	DOCENTE	ESTUDIANTE
Tradicional	El aprendizaje se da por repetición, al pie de la letra y no se motiva a que los estudiantes reflexionen por lo aprendido	Es el poseedor del conocimiento, es el centro de atención durante la clase. Establece normas y las hace cumplir.	Su papel es pasivo y debe obedecer todo lo que se le dice y acatar las normas del colegio.
Conductista	Las estrategias de enseñanza parten de objetivos, los contenidos se transmiten utilizando medios didácticos pero la evaluación es de forma memorística y cuantitativa.	Guía al estudiante hacia el logro de un objetivo instruccional. El plan de enseñanza son los objetivos, las experiencias educativas, su organización y su evaluación. Relación Maestro – Alumno : Intermediario	El modelo por objetivos tiende a sistematizar, medir, prever, evaluar, clasificar y proyectar cómo se va a comportar el alumno después de la instrucción.
Cognoscitivista	Enfatiza la importancia de la experiencia en el desarrollo de los procesos cognitivos. El sujeto tiene carácter activo en sus procesos de conocimiento y de desarrollo cognitivo.	El rol del docente es tener en cuenta el nivel de desarrollo y el proceso cognitivo de los alumnos. Orienta a los estudiantes a desarrollar aprendizajes por recepción significativa y a participar en actividades exploratorias, que puedan ser usadas posteriormente en formas de pensar independiente.	Puede contribuir de diversas maneras a lograr el aprendizaje significativo.
Socialista	Se da preferencia a la autoevaluación y coevaluación, pues el trabajo es principalmente solidario.	Es un investigador de su práctica y el aula es un taller	Desarrollan su personalidad y sus capacidades cognitivas en torno a las necesidades sociales para una colectividad en consideración del hacer científico.
Constructivista	Se forman sujetos activos, capaces de tomar decisiones y emitir juicios de valor, lo que implica la participación activa de profesores y el estudiante	Es un facilitador que contribuye al desarrollo de capacidades de los estudiantes para pensar, idear, crear y reflexionar.	Interactúan en el desarrollo de la clase para construir, crear, facilitar, liberar, preguntar, criticar y reflexionar sobre la comprensión de las estructuras profundas del conocimiento

MODELO	EDUCACIÓN	DOCENTE	ESTUDIANTE
Conceptual	Busca formar instrumentos de conocimiento, desarrollando las operaciones intelectuales y privilegiando los aprendizajes de carácter general y abstracto sobre los particulares	Promueve el pensamiento, las habilidades y los valores en sus educandos, diferenciando a sus alumnos según el tipo de pensamiento por el cual atraviesan (y su edad mental), y actuando de manera consecuente con esto, garantizando además su aprendizaje de los conceptos básicos de las ciencias y las relaciones entre ellos.	Organizan sus ideas y las comparten, utilizando elementos que les permita pasar de lo abstracto a lo particular
Tecnología Educativa NTIC	Implementar estrategias didácticas activas que faciliten el aprendizaje autónomo, colaborativo y el pensamiento crítico y creativo mediante el uso de las TIC, y, diseñar currículos colectivamente con base en la investigación	Es importante resaltar el proceso de cualificación en la formación docente, en particular en uso y apropiación de las TIC y la importancia de fortalecer los planes de estudio que respondan a las necesidades específicas de las comunidades.	Realizan un aprendizaje individual con pensamiento crítico y creativo. Interactúan con el medio en forma colaborativa.

Actualmente, acerca de la Tecnológica siguen existiendo grandes diferencias y contradicciones con respecto a cómo incorporarla en el aula de clase. Esto se ha sustentado en la afirmación de que no existen teorías completas que orienten este tipo de práctica educativa y mucho menos que la expliquen totalmente. Es así como cada Proyecto de Tecnología Educativa adopta su propio “Modelo Pedagógico”.

El Modelo Pedagógico para nuestro Proyecto, está construido sobre la base de una concepción de interactividad cognitiva, diferenciada de la mera interactividad instrumental. Nos situamos en la convicción de que los medios pueden favorecer la interactividad cognitiva, pero este proceso no se establece por sí mismo ni de manera instrumental. El desafío es (previa capacitación de los formadores), diseñar una serie de acciones didácticas, secuenciadas y ordenadas que le permitan al alumno la reconstrucción significativa de los contenidos en la medida en que va reconstruyendo su mirada del mundo y de sí mismo.

La Tecnología en las escuelas

Empecemos por diferenciar dos términos muy comunes, ya definidos en la Introducción: la **Educación Tecnológica** y la **Tecnología Educativa**, entendiendo a la primera como una forma de conocimiento escolarizado y a la segunda como un dispositivo dentro de un sistema tecnológico más amplio: el educativo, que contiene a la educación escolar como tecnología social.

DIFERENCIAS	
Educación Tecnológica	Tecnología Educativa
Objetivo: Formar una cultura tecnológica	Objetivo: Satisfacer las demandas sociales
Contexto: escolar	Contexto: socio-económico-productivo
Fin: Formar sujetos críticos y reflexivos que actúen en el medio artificial	Fin: Fabricar productos tecnológicos

Educación Tecnológica

En nuestro país, durante mucho tiempo el propósito de la Educación Técnica ha sido la preparación de técnicos para abastecer el sistema industrial o la formación de oficios, es decir, orientado hacia el mundo del trabajo. Lo importante era darles conocimientos que les permitiera ser operarios con las habilidades y destrezas que se requerían en ese momento. El técnico especializado era necesario, más allá del operario de baja calificación de la línea de producción.



En toda empresa que elabora productos tecnológicos coexisten tecnologías duras y tecnologías blandas. Así, por ejemplo, en una fábrica de automóviles hay un importante desarrollo de tecnologías duras destinadas a producir vehículos y paralelamente es indispensable el desarrollo de tecnologías gestionables destinadas a la organización, la administración, la publicidad y la venta del producto.

Entonces, la Tecnología es una actividad social centrada en el saber hacer, donde se forman los profesionales que elaboran diseños, ejecutan proyectos, organizan y supervisan la organización industrial.

El Modelo Pedagógico está orientado a lograr un crecimiento técnico que genere un futuro mejor y que requiere la formación de gente entrenada para estos propósitos. La relación alumno/profesor es con vínculo de dependencia.

En las últimas tres o cuatro décadas, en Argentina, se produjeron modificaciones de significativa magnitud y cuantía; por un lado, importantes cambios tecnológicos que

influyeron fuertemente en el modo de vida de toda la sociedad y, por otro, un marcado proceso de desindustrialización a nivel país.

En esta situación, nuestra Educación Técnica seguía formando Técnicos e Ingenieros, pero en un marco de proyección futura cada vez más incierto e inseguro, entonces, se entra a un estado de duda y de confusión, tanto en lo que hace a la educación como en lo que respecta a la sociedad, a la que le cuesta construir escenarios de futuro previsible.

En este contexto, desde distintos sectores, comienzan a pensarse que es necesaria la presencia del **Área de Tecnología en el Sistema Educativo** como parte de la formación general del ciudadano. En esto, es probable que haya influido lo ocurrido en los sistemas educativos de otros países, pero también el desarrollo de corrientes internas de pensamiento que se fueron postulando, a saber:

1. Desde los sectores técnicos, ingenieriles y científicos, se percibe que la concentración del conocimiento y de la producción tecnológica inclina cada vez más la balanza del poder y, por lo tanto, no puede existir un proyecto de país en un mundo cada vez más globalizado y dentro de un sistema capitalista, si no hay producción tecnológica y científica. Estos sectores consideran imperioso realizar algo en este sentido, como ser la incorporación de Tecnología en la formación general de los alumnos.
2. Distintos círculos de pensadores y educadores comienzan a pensar que si la Tecnología está influyendo tan fuertemente en la realidad, es necesario que a nivel de ciudadanos se pueda entender: qué es la tecnología, cómo se produce, qué significa, cómo y para qué funciona, qué rol le compete a los distintos grupos sociales.
3. La Tecnología cada vez se interrelaciona más con la ciencia y se potencian mutuamente, lo cual lleva a una tercera línea de pensamiento que intenta reflexionar sobre la relación entre técnica, ciencia y sociedad.
4. Hay quienes sostienen que hubo una cuarta corriente que propició la incorporación de tecnología dentro del sistema educativo, aquella que la considera necesaria para la generación de mano de obra calificada de acuerdo con los requerimientos actuales. Al respecto, conviene aclarar que si bien existe una relación entre lo que se aprende en la escuela y el mundo del trabajo y que las habilidades que allí se desarrollan posicionan al individuo de diferente manera, es claro que la preparación de mano de obra especializada no puede formar parte de la formación de todos los educandos.

Es razonable suponer que estas corrientes de pensamiento son las que confluyeron en la idea de que es necesario incluir el Área de Tecnología en el Sistema Educativo. Así, desde hace más de diez años en la Argentina comenzó a implementarse la Educación Tecnológica como área de formación general.

Tecnología Educativa

Los orígenes de la Tecnología Educativa pueden hallarse en la enseñanza programada, con la idea de elevar la eficiencia de la dirección del proceso docente.

Su creación se debe a B. F. Skinner, profesor de la Universidad de Harvard, 1954. Sus trabajos se enmarcan en la corriente psicológica del conductismo, la que considera el aprendizaje básicamente en la fijación de un repertorio de estímulos del medio y sus respuestas (E - R). Este modelo psicológico del aprendizaje sirvió de base para la enseñanza programada, primera expresión de la Tecnología Educativa.

El Modelo Pedagógico presente en esta tendencia se puede resumir en:

- ➔ Objetivos conductuales,
- ➔ Organización del contenido de forma lógica en secuencia de unidades;
- ➔ Métodos basados en el autoaprendizaje para lo que se utilizan las preguntas y respuestas,
- ➔ Utilizan los juegos didácticos y las simulaciones; y los medios docentes son libros, máquinas de enseñar, computadoras y TV.
- ➔ La relación alumno/profesor prácticamente no existe; el profesor elabora el programa y el alumno se autoinstruye, a su ritmo, despersonalizándose el proceso docente, eliminándose su influencia educativo – formativa.



Esta corriente pedagógica ha sido ampliamente difundida en América Latina a través de la influencia del sistema norteamericano de enseñanza. Sus seguidores le reconocen las siguientes **ventajas**:

- ➔ la constante activación de los alumnos,
- ➔ la individualización del aprendizaje,
- ➔ la comprobación directa y
- ➔ la corrección de los resultados instructivos.

No caben dudas que la masividad de la enseñanza y la educación a distancia, encuentran en la enseñanza programada sus requerimientos satisfechos. No obstante presentan algunas **limitaciones** como las siguientes:

- En el aprendizaje no se toman en cuenta los procesos ni las cualidades, sino los resultados instructivos.
- La orientación de las acciones del alumno es, generalmente, por ensayo y error.
- No desarrolla el pensamiento teórico, ni creador, sino la memoria reproductiva.

Sin aplicarse exactamente a la enseñanza programada, el pensamiento crítico y creativo del modelo tecnológico de enseñanza, apoyado en el aprendizaje autónomo y colaborativo, ha tomado cuerpo en el carácter instrumental de algunas didácticas y en muchas prácticas docentes. En efecto, la educación aparece descontextualizada, sin tener en cuenta sus realidades y conflictos, y el proceso se centra en lo que puede ser controlado.

Rol del docente	Rol del estudiante	Características de la clase
<ul style="list-style-type: none"> • Selecciona tareas y respuestas. • Elabora un programa de enseñanza. 	<ul style="list-style-type: none"> • Aprendizaje individualizado. • Papel preponderante. • Se autoinstruye. • Aprende de acuerdo a su ritmo individual de asimilación por ensayo-error. • Desarrolla memoria reproductiva, favorece pensamiento teórico creativo 	<ul style="list-style-type: none"> • La atención se dirige a métodos y medios más que a contenidos (tecnología de la instrucción). • Se basa en Estímulo - Respuesta. Estímulo - Conducta - Reforzamiento. • Enseñanza programada mediante máquinas de enseñar (programación del aprendizaje). Es un programa lo que el alumno "toca" y recibe información del resultado de su actividad. • Objetivo conductual. Organización lógica por unidades del contenido. • Método de autoaprendizaje. Medios: TV, libros, máquina. • Evalúa resultados y no proceso. • Actualmente se usa como medio que facilita la enseñanza p ej. en educación a distancia, pero se utilizan todas las bondades de las nuevas tecnologías de la información y las comunicaciones (NTIC)

Acercar la Programación a las escuelas

“El hombre ha intentado con gran esfuerzo reproducir habilidades mentales en máquinas y lo ha logrado a una escala pequeña.”

Alan Turing (1912-1954) es considerado uno de los padres de la ciencia de la computación y precursor de la informática moderna. Proporcionó una influyente formalización de los conceptos de algoritmo y computación: la máquina de Turing. En el campo de la inteligencia artificial es conocido sobre todo por la concepción del test de Turing, un criterio según el cual puede juzgarse la inteligencia de una máquina, si sus respuestas en la prueba son indistinguibles de las de un ser humano.

Hoy en día, se puede ver la inteligencia artificial en varios lugares a nuestro alrededor, el más cercano son los videos juegos que simulan pensar por sí solos.

Con los trabajos de Turing se creó una nueva disciplina y con ésta se abrió la mente del ser humano, porque no solo ha ideado máquinas inteligentes sino que también utiliza el término para concebir películas de ciencia ficción donde las máquinas llegan a tener una inteligencia similar a la de los seres humanos, ejemplo de ello son las películas como “IA”, o “Yo Robot”, donde explotan el tema de las máquinas inteligentes.

Consideremos que en la actualidad, a nivel mundial, las tecnologías digitales constituyen parte de la cultura social de los jóvenes y adolescentes, de sus modos de vincularse y de representarse en el mundo. Nuestros jóvenes son “nativos digitales”; en sus prácticas está naturalizado el envío de mensajes de texto, el uso de los juegos en línea, navegar en Internet, etcétera. Sin embargo, a pesar de interactuar todo el tiempo con medios digitales, muy pocos de nuestros jóvenes pueden construir nuevos contenidos digitales como podrían ser sus propios juegos, animaciones y simulaciones.

En este Proyecto realizamos una **investigación** con el objetivo de *buscar una alternativa diferente para acercar a las escuelas secundarias la enseñanza de Programación*, atendiendo el rol central que ocupan los videojuegos en la vida cotidiana de los jóvenes y a la Programación como habilidad requerida para poder crear con los medios digitales. De este modo también podremos incentivar el estudio de las carreras de Ingeniería en Informática, Sistemas y Computación.

Nuestra investigación se centró en los siguientes ejes:

- **La currícula de los colegios secundarios**, está en constante cambio y las Tecnologías de la Información y Comunicación (TICs) se han incorporado de diferentes maneras en la escuela a lo largo de los últimos años. En relación a la enseñanza de Programación, la situación es de lo más dispar, cada Institución lo hace de acuerdo a la orientación del docente a cargo y de la formación del mismo, algunas escuelas se orientan hacia el uso de TICs para la comunicación, otras hacia la ofimática y otras incorporan

lenguajes de programación generando una situación compleja en cuanto a los alcances y tipos de enseñanzas.

➔ **Los docentes que deben promover y facilitar el aprendizaje de sus alumnos**, la primera cualidad que debe demostrar es el dominio y la maestría en su área o disciplina. Sin embargo, además del dominio temático especializado, un docente requiere de la capacidad de llevar esos conocimientos a situaciones reales de enseñanza-aprendizaje con sus estudiantes, lo cual implica la puesta en práctica de creatividad y pensamiento crítico, para desarrollar prácticas pedagógicas (de planeación, didácticas y evaluativas).

➔ **Los estudiantes** que poseen plenas habilidades para el uso de múltiples herramientas digitales, pero no logran adquirir las competencias necesarias para crear contenidos propios a partir de ellas.

En consecuencia, en una primera etapa, analizamos los espacios curriculares en los que se podría intervenir con los temas de Programación Orientados a Objetos (POO) y al lenguaje JAVA. Advertimos, la necesidad de trabajar con un material didáctico más cercano a la cultura adolescente que nos permita aproximarnos a la Programación en JAVA desde una perspectiva diferente, con un abordaje más acrecentado y con estrategias lúdicas que promuevan competencias. En ese sentido se incorpora al Proyecto la herramienta didáctica de Robocode.

Tomamos como nivel inicial para implementar los contenidos de programación, 3° año del ciclo básico, ya que es en el transcurso de este año donde los estudiantes eligen con qué Especialización continuarán. Dentro de nuestra propuesta, el "**Taller de Programación**" es el espacio curricular elegido para el desarrollo de estas nuevas estrategias y herramientas didácticas para la enseñanza de Programación.

Luego, observamos que es necesaria la "actualización docente" en el área de programación, los profesores al enfrentarse a estos nuevos escenarios educativos (lenguaje Java y Software Robocode), deben estar preparados para clarificar los contenidos, estimular el aprendizaje, incentivar la creatividad y evaluar el progreso de los alumnos, ya que son ellos quienes deben asumir las funciones orientadoras, cognoscitivas, motivadoras y de nexos dentro del espacio de instrucción del Taller de Programación. Por esta razón también **implementaremos las "Jornadas de Capacitación de Capacitadores"**.

Finalmente advertimos que los estudiantes, con su avidez adolescente por todo lo tecnológico y lúdico, estarían dispuestos, en su mayoría, a realizar el Taller de Programación para poder participar de las Olimpiadas de Robocode. Como ya dijimos con este Taller acercamos la Universidad a la Escuela, pues los alumnos al finalizar habrán logrado las competencias necesarias para crear, explorar, solucionar problema y entre otras, elaborar estrategias (en nuestro caso, de combate entre robots virtuales en el campo de batalla de Robocode).

Estrategias Metodológicas

De acuerdo con lo planteado hasta ahora en nuestro Proyecto, tomaremos las siguientes definiciones para fortalecer nuestras estrategias de enseñanza-aprendizaje:



1°

Estrategias Metodológicas:

Hacen referencia a la forma de enseñanza, al cómo se enseña, a los caminos que sigue el docente para conseguir que los alumnos aprendan, a las vías que orientan la enseñanza para el logro de los objetivos de enseñanza planeados.

2°

Tecnología para Programación:

Son los elementos tecnológicos necesarios (lenguaje de programación, lenguaje de modelaje, etc.) para expresar, en un lenguaje comprensible por una máquina, la abstracción de la realidad y el algoritmo que resuelve un problema sobre dicha abstracción.

Programar es la habilidad de utilizar dicha tecnología para que una máquina sea capaz de resolver el problema.

3°

Herramientas de Programación:

Son las herramientas computacionales (compiladores, editores, depuradores, manejadores de proyectos, etc.) que permiten a una persona desarrollar un programa. Se pueden considerar una implementación particular de la tecnología.

1°

Aplicando la primera definición, para lograr un adecuado proceso de enseñanza-aprendizaje en nuestro Programa Incentivo de Programación, generaremos un espacio físico denominado **Aula-Taller**, donde se realizarán las actividades necesarias para que los **Alumnos**, a través de los **Contenidos** (incluidos Herramientas y Tecnología) y guiados por el **Docente**, alcancen un adecuado nivel en Programación.

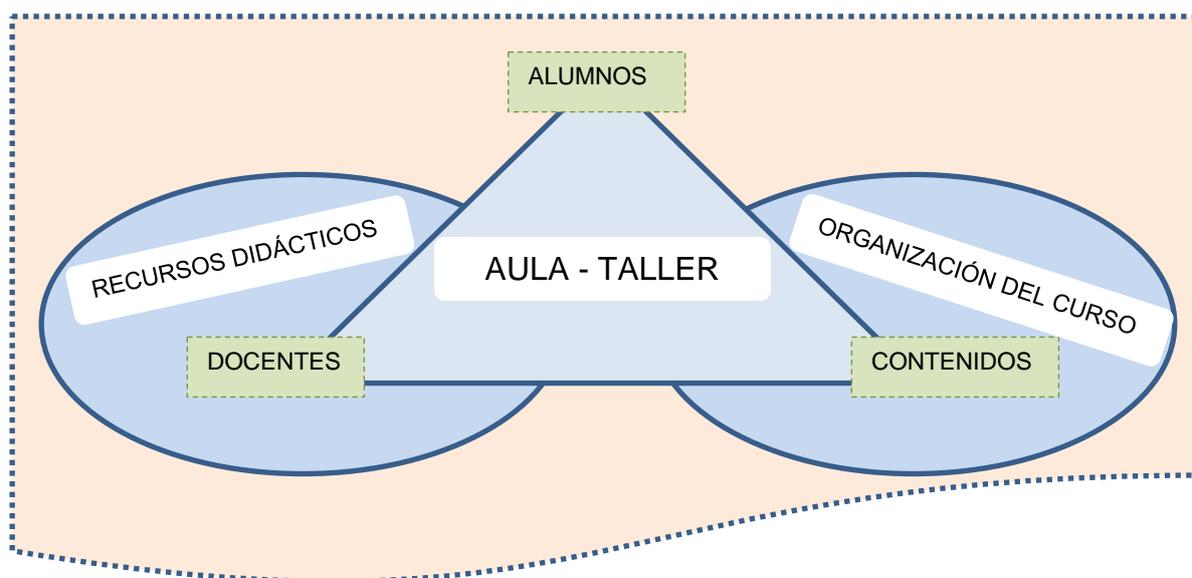


Gráfico 1

Estrategias Metodológicas para el Programa Incentivo de Programación:

-  **El aula-taller:** Se debe organizar en concordancia con la institución educativa considerando que debe ser un espacio de trabajo, con sillas, mesas, computadora y conexión a Internet (herramientas indispensables para el proceso de aprendizaje de los alumnos), un espacio en el que resulte agradable estar y hacer cosas, que permita la participación activa de todos, donde cada uno construya su conocimiento en interacción con los demás. Un ámbito en el que interactúan la acción y la reflexión.
-  **El docente:** Después de realizar las Jornadas de Capacitación de Capacitadores debe: diseñar: la estrategia, las actividades de motivación, el trabajo: (en grupo, individual, de reflexión, de puesta en común, de laboratorios, de entrenamiento, etc.), También deberá evaluar formativamente el progreso, para que el estudiante tenga siempre información de qué está haciendo bien y qué debe corregir.
-  **El alumno:** Tiene un papel preponderante y un aprendizaje individualizado. Se autoinstruye. Aprende de acuerdo a su ritmo individual de asimilación por ensayo error. Desarrolla memoria reproductiva. Proceso de aprendizaje por imitación. Pensamiento teórico creativo.
-  **Los Contenidos:** Para hacerlos más comprensibles incluimos ejemplos y casos concretos. Insertamos lecturas y vínculos que supongan una ayuda adicional a los contenidos presentados. Realizamos contenidos accesibles para todos los alumnos, que puedan ser recopilados y permitan su fácil lectura y estudio.

 **Los Recursos Didácticos:** Usamos los recursos multimedia que apoyan al contenido textual, usamos la: Plataforma gratuita de Robocode, Tutoriales, Entrenadores, Simuladores, Frameworks

 **Organización del curso:** Al diseñar el curso tuvimos en cuenta la forma de generar las habilidades, cómo manejar la motivación y la frustración, el proceso de aprendizaje y el uso de los recursos disponibles, los ejercicios, la solución de problemas y finalmente la evaluación.

2° Considerando lo especificado en la segunda definición, filtramos (de todos los datos recaudados desde el título “Aspecto Técnico”: Java y Robocode), aquella información que resulta básica para generar los conocimientos necesarios para que alumnos de escuela media puedan desarrollar sus robots, estimulando así, habilidades tales como la lógica, la matemática, el diseño, la solución de problemas y la creatividad.

3° Las herramientas de programación, que utilizamos son aquellas que permiten realizar aplicativos, programas, rutinas, utilitarios y sistemas para que la parte física del ordenador, funcione y pueda producir resultados. Entre otras herramientas de programación usaremos bibliotecas y componentes, dados por el lenguaje de programación Java.

Resumiendo, podemos decir que el elemento que orientó el diseño de nuestra Estrategia Metodológica, es la generación de ambientes de aprendizaje lúdicos, donde la curiosidad por el funcionamiento de las cosas, la indagación de fenómenos, la exploración de ideas, soluciones, sitios o eventos cotidianos, la experimentación y la manipulación de artefactos, procesos y herramientas, son los derroteros de acción.

Es en este tipo de ambiente de aprendizaje donde el docente capacitado y actualizado en el tema, desarrolla las diferentes situaciones didácticas diseñadas y propone diferentes retos en compañía de sus estudiantes.

La incorporación de una manera planificada y reflexiva de nuevas herramientas didácticas relacionadas al aprendizaje de la Programación en la escuela secundaria, promueve habilidades que son útiles para otras áreas del conocimiento, amplía las posibilidades de las “cosas” que se pueden crear, inventar con la computadora y en general de las cosas que se pueden aprender.

Entonces, ¿por qué no enseñarles a los alumnos secundarios a crear sus propias herramientas para expresar sus ideas? Solo necesitan aprender las bases de la Programación y así convertirse en agentes creativos capaces de diseñar y desarrollar sus propias tecnologías.

Instrumentos: Elementos de la Programación Básica

Sintaxis:

Cada lenguaje de programación tiene unas reglas especiales para la construcción de programación, a esto se le denomina sintaxis.

El compilador lee el programa y comprueba que el programa sigue las reglas de sintaxis del lenguaje de programación, el compilador traduce el código fuente de Java a un código máquina (código objeto).

Código Objeto:

Consta de instrucciones máquina e información de cómo cargar el programa en la memoria antes de su ejecución.

Si el compilador encuentra errores, los presentará en la pantalla, una vez corregidos los errores se vuelve a compilar sucesivamente hasta que no se produzcan más errores.

Depuración:

Los programas rara vez funcionan bien la primera vez que se ejecutan, por lo que los errores que se detectan deben ser corregidos.

Al proceso de encontrar errores se denomina depuración del programa, esta tarea es de las más difíciles en el proceso de programación.

Errores de sintaxis:

Son aquellos que se producen cuando el programa viola la sintaxis, es decir, las reglas gramaticales del lenguaje.

Errores lógicos:

Son errores del programador en el diseño del algoritmo, son difíciles de encontrar y aislar, ya que no suelen ser detectados por el compilador.

Errores de regresión:

Son aquellos que son provocados cuando accidentalmente se producen al corregir un error lógico y se produce otro en otra parte del programa.

Elementos léxicos de los programas:

- Identificadores
- Palabras reservadas
- Literales
- Operadores
- Separadores

Variable:

Es un espacio en la memoria de la computadora que permite almacenar temporalmente un dato durante la ejecución de un proceso, su contenido puede cambiar durante la ejecución del programa.

Para poder reconocer una variable en la memoria de la computadora, es necesario darle un nombre con el cual podamos identificarla dentro de un algoritmo.

Identificador:

Son nombres que el programador asigna a: variables, constantes, clases, métodos, paquetes, etc. de un programa.

Características:

- Están formados por letras y dígitos
- No pueden empezar con un dígito
- No pueden contener ningún carácter especial (solo signo \$).

Ejemplo:

Edad
nombre
_Precio
\$cantidad
PrecioVentaPublico

Palabras Reservadas:

Es aquella que tiene un significado especial para el lenguaje de programación.

Una palabra reservada no puede ser utilizada como identificador, objeto o función:

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

Signos de Puntuación y Separadores:

Todas las sentencias deben de terminar con un punto y coma (;).

Otros signos de puntuación son:

+ - * / = % & # ! ? ^ " ' ~ \ | < > () [] { } : ; , .

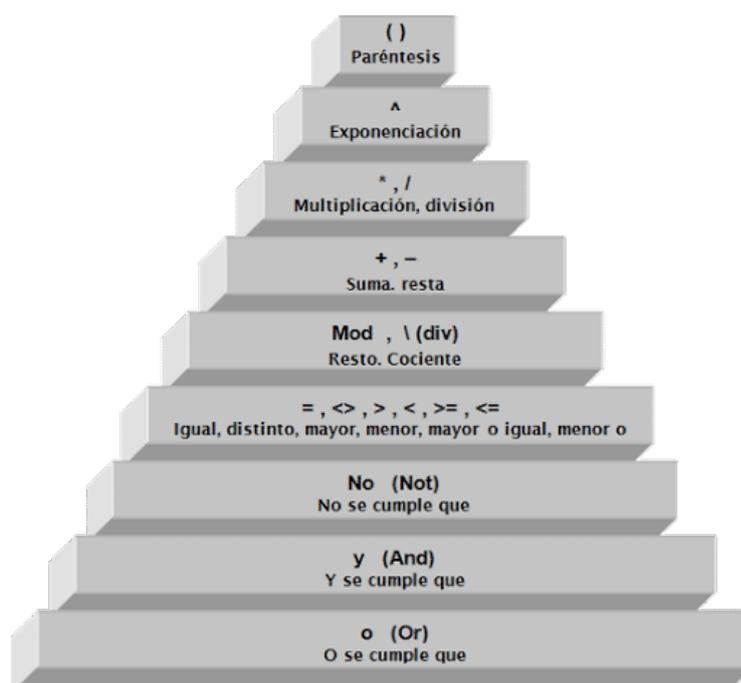
Operadores en Java:

Operador	Descripción
<code>==</code>	Es igual
<code>!=</code>	Es distinto
<code><</code> , <code><=</code> , <code>></code> , <code>>=</code>	Menor, menor o igual, mayor, mayor o igual
<code>&&</code>	Operador and (y)
<code> </code>	Operador or (o)
<code>!</code>	Operador not (no)

Prioridad de Operadores Aritméticos:

Todas las expresiones con paréntesis anidados se evalúan de dentro a afuera, el paréntesis más interno se evalúa primero.

Dentro de una misma expresión los operadores se evalúan en el siguiente orden:



Los Operadores en la misma expresión con igual nivel de prioridad, se evalúan de izquierda a derecha.

Algebra:

Estamos acostumbrados a representar algebraicamente una ecuación, pero en la computadora es diferente, para ello podemos utilizar los paréntesis, de lo contrario el resultado puede ser diferente.

Constante:

Es un dato numérico o alfanumérico que no cambia durante la ejecución del programa. Ejemplo: $\pi = 3.1416$

Ejemplo:

$\text{área} = \text{Math.PI} * \text{radio}^2$ Las variables son : el radio, el área y la constante es PI

Aspectos Técnicos de Java y Robocode

Java



Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener pocas dependencias de implementación como fuera posible.

Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.

Filosofías de Java

El lenguaje se creó con cinco objetivos principales:

1. Debería usar el paradigma de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.

5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Orientado a objetos: Se refiere a un método de programación y al diseño del lenguaje. Una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos.

Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos). El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos.

Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software.

Independencia de plataforma: Significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo.

Funcionamiento: Se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode, instrucciones máquina simplificadas específicas de la plataforma Java). Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino.

El bytecode es ejecutado entonces en la máquina virtual (JVM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hilos o threads, la interfaz de red) de forma unificada.

Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT (Just In Time).

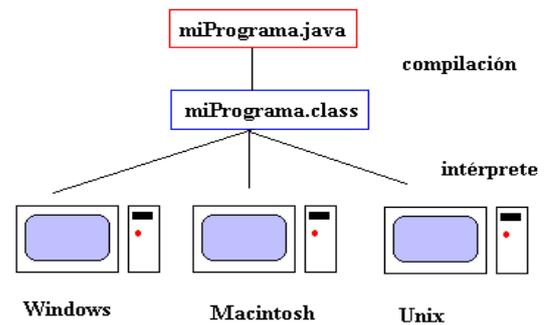
Bibliotecas de Clases Java

Los programas en Java constan de varias piezas llamadas “Clases”.

Estas “Clases” incluyen componentes llamados “Métodos” los cuales realizan las tareas y devuelven información cuando completan esas tareas.

Los programadores pueden crear cada uno de los componentes que necesitan para formar programas en Java. Pero la mayoría, aprovechan las llamadas Bibliotecas de Clases Java (APIs).

Una Biblioteca es una Clase Java definida, que puede ser utilizada en la Clase que es llamada. Esto es, puedo utilizar los métodos declarados en la Clase importada, en el programa que se está desarrollando.



Entorno de desarrollo típico en Java

Por lo general, los programas Java pasan a través de 5 fases:

1. Edición (*.java)
2. Compilación (*.class)
3. Carga
4. Verificación
5. Ejecución

Variables

Una variable es un espacio de memoria y un nombre simbólico de ese espacio de memoria. Las variables pueden ser:

- Del tipo int (números naturales),
- Del tipo float (números de punto flotante),
- Del tipo char (caracteres).

Las variables dependen en donde son declaradas:

- ▶ Si son declaradas dentro de un Método, solo funcionan dentro de ese Método.
- ▶ Si son declaradas fuera de un Método pero dentro de la cCase son las llamadas variables globales.

Métodos

- ✓ Un método es un bloque de construcción del programa.
- ✓ Un método puede recibir argumentos, esto es valores que el método va a utilizar.
- ✓ Un método puede o no devolver un valor.

Operadores

Los operadores en Java los podemos distinguir en aritméticos y lógicos:

➡ Operadores Aritméticos:

➡ Suma, Resta, Multiplicación, División:

+ , **-** , **X** , **/**

➡ Módulo: **%**

Ej. $23 \% 10 = 3$

➡ Operadores Lógicos:

➡ AND: **&&** . Ambas condiciones deben ser verdadero

➡ OR: **||** . Por lo menos una de las condiciones verdadero

➡ Negación: **~** . Negar la condición

Algoritmos

Cualquier problema de computación puede resolverse ejecutando una serie de acciones en un orden específico. Un procedimiento para resolver un problema en término de:

➡ las acciones a ejecutar

y

➡ el orden en el que se ejecutan estas acciones

se conoce como algoritmo.

Al proceso de especificar el orden en el que se ejecutan las instrucciones (acciones) en un programa, se le llama **Control de Programas**.

Estructuras de control

Generalmente, en un programa las instrucciones se ejecutan una después de otra, en el orden en que están escritas. Este proceso se conoce como **ejecución secuencial**.

Varias instrucciones en Java permiten al programador especificar que, la siguiente instrucción a ejecutarse, tal vez no sea la siguiente en secuencia. Esto se conoce como **transferencia de control**.

Instrucciones de selección en Java

Java tiene tres tipos de instrucciones de selección:

- ➔ La instrucción **if** realiza (selecciona) una acción si la condición es verdadera, o evita la acción si la condición es falsa.
- ➔ La instrucción **if...else** realiza una acción si la condición es verdadera, o realiza una acción distinta si la condición es falsa.
- ➔ La instrucción **switch** realiza una de entre varias acciones distintas, dependiendo del valor de una expresión.

Instrucciones de repetición en Java

Java cuenta con tres instrucciones de repetición (también llamadas instrucciones de ciclo) que permiten a los programas ejecutar instrucciones en forma repetida, siempre y cuando una condición (condición de continuación de ciclo) siga siendo verdadera.

Las instrucciones de repetición se implementan con las instrucciones **while**, **do...while** y **for**.

- ➔ Las instrucciones **while** y **for** realizan la acción (o grupo de acciones) en sus cuerpos, cero o más veces; si la condición de continuación del ciclo es inicialmente false, no se ejecutara la acción (o grupo de acciones).
- ➔ La instrucción **do...while** realiza la acción (o grupo de acciones) en su cuerpo, una o más veces.



Robocode

Robocode es un juego educativo *Open Source* escrito en *Java*, en el que los competidores deben diseñar y programar sistemas de control para unos tanques (robots), que compiten con el objetivo de eliminar el resto de los adversarios.

Robocode, desarrollado por IBM, es un simulador de tanques de combate, en el que deben recorrer el entorno para evitar ser disparados por sus oponentes y evitar también chocar contra las paredes. Además, para ganar, el tanque debe localizar a su adversario y dispararle.

Simulador:

Herramienta interactiva que permite capacitar y entrenar a los alumnos en un entorno muy similar al real.

De esta forma los alumnos tienen la posibilidad de tomar decisiones, analizar opciones, ejecutar procedimientos y equivocarse la cantidad de veces que sea necesario para aprender.

Como todos los simuladores, Robocode recoge algunos aspectos de la realidad, pero obvia otros:

Aspectos de la realidad reflejados en el juego:

- Las actuaciones llevan tiempo, mientras rotas te pueden disparar.
- Las municiones tardan en llegar. Si disparas a un objetivo que se mueve, debes tener en cuenta donde estará cuando la munición llegue.
- El sensor unidireccional debe apuntar al adversario para verlo.
- El cañón se calienta al disparar y debe enfriarse antes del próximo disparo.
- Chocar contra los obstáculos debilita.

Aspectos no realistas en el juego:

- Los sensores y el resto de los componentes son silenciosos.
- Los sensores detectan velocidad, posición, orientación y energía restante.
- Los combates tienen lugar en un entorno plano (2D) cerrado.

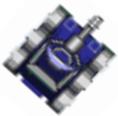
Facilidades que brinda Robocode

Entre los motivos ya mencionados sobre el uso de la herramienta de Robocode, también podemos enumerar:

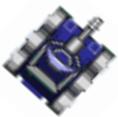
1. La posibilidad de utilizar un editor de texto para escribir el comportamiento del robot.
2. Compatibilidad con IDE de programación como lo son Eclipse y NetBeans.
3. Un campo de batalla para llevar a cabo la simulación de los tanques.

El Robot

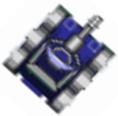
Un robot es un tanque formado por:



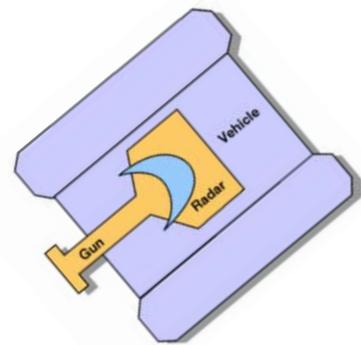
Cuerpo (body/vehicle): Lleva encima el arma con el radar. Los movimientos que puede hacer el cuerpo son hacia adelante, hacia atrás, hacia la izquierda o hacia la derecha.



Arma (gun): Montada sobre el cuerpo, es usada para disparar las municiones. Los movimientos que puede hacer son hacia la izquierda o hacia la derecha.



Radar: Montado sobre el arma, es usado para escanear (encontrar) otros robots mientras se mueva. El movimiento que puede realizar el radar es, hacia la izquierda o hacia la derecha. Este radar genera una señal cuando un robot enemigo es detectado.



Reglas del Juego

Los tanques en Robocode se dirigen por eventos, hay un bucle principal, que puede ser interrumpido por un número de manejadores de eventos que incluyen acciones como:

- ▶▶ Ver al adversario.
- ▶▶ Ser alcanzado por una munición.
- ▶▶ Chocar con un adversario

Los tanques comienzan con una determinada cantidad de energía, que se va perdiendo por:

- ▶▶ Recibir un disparo.
- ▶▶ Chocar contra la pared o un adversario.
- ▶▶ Por disparar municiones.

También, si el juego dura mucho tiempo, todos los robots van perdiendo energía hasta que uno de ellos muere. Esto impide que la partida dure infinitamente.

Cuando dos robots colisionan, ambos pierden la misma cantidad de energía, por eso, chocar puede ser una buena estrategia para robots que están a punto de ganar.

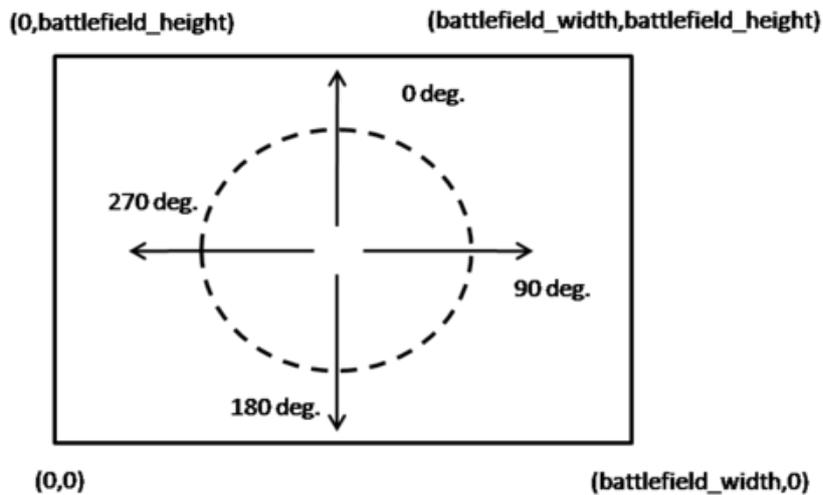
Sólo se puede ganar energía acertando los disparos a los adversarios.

Si un robot se queda sin energía debido a su tiroteo, se inhabilita. Pero si una munición da a su enemigo después, algo de energía le devuelve. Si un robot se queda sin energía por alguna razón (como ser alcanzado por una munición o por chocar contra la pared) entonces, muere.

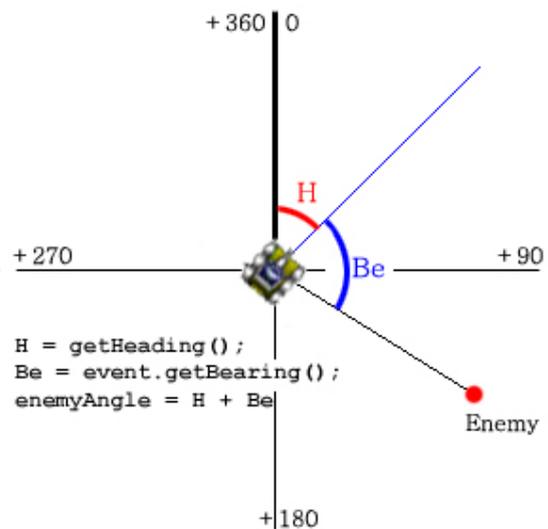
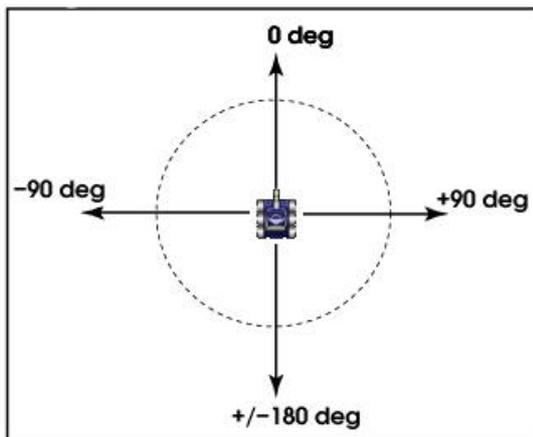
Aspectos a tener en cuenta

Robocode utiliza el sistema de coordenadas cartesianas, es decir, la coordenada (0,0) está ubicada abajo a la izquierda del campo de batalla.

La dirección u orientación está ubicado como las agujas del reloj: 0/360 grados hacia el "Norte", 90 grados hacia el "Este", 180 grados hacia el "Sur", y 270 grados hacia el "Oeste".



En el bearing (ángulo relativo al cuerpo del robot), el rango de grado va desde los -180 a 180 grados. El mismo, es relativo a la posición en grados a la orientación de nuestro robot.



Energía

Cada robot comienza con 100 unidades de energía. La misma desciende por:

- ➡ Realizar un disparo.
- ➡ Es alcanzado por un disparo de un robot enemigo.
- ➡ Colisiona contra otro robot.

Y asciende cuando:

➔ Un disparo propio alcanza a un enemigo.

Disparo

Cada disparo tiene asociada una energía (ED). De dicha energía depende la velocidad del proyectil:

$$\text{Velocidad del Proyectil} = 20 - 3 * ED$$

El daño al oponente depende de la energía utilizada:

$$\text{❖ Daño} = 4 * ED, \text{ si } ED \leq 1$$

$$\text{❖ Daño} = 4 * ED + 2 * (ED - 1), \text{ si } ED > 1$$

Recuperación de energía en caso de acierto:

$$\text{Recuperación} = 3 * ED$$

Calentamiento del cañón en cada disparo (no se puede disparar si el cañón está caliente):

$$\text{Calentamiento} = 1 + ED$$

Estrategias de radar:

- 🔊 **Simple:** Apuntar el radar en la misma dirección que el arma.
- 🔊 **Circular:** Mantener el radar girando 360 grados constantemente.
- 🔊 **Tracking:** Mantener el radar fijo sobre un objetivo.
- 🔊 **Optimal:** Mover el radar de modo de escanear todos los robots tan rápido como sea posible.

Estrategias de movimiento:

- 📏 **Movimiento en línea recta.**
- 📏 **Movimiento circular:** El robot describe un círculo en el campo de batalla.
- 📏 **Movimiento oscilante hacia adelante y hacia atrás.**
- 📏 **Anti-gravedad:** mantenerse lejos de áreas peligrosas

Estrategias de disparo:

- ★ **Estacionario:** Dispara hacia la actual ubicación del objetivo. Es la más simple y menos efectiva de las estrategias.
- ★ **Lineal:** Dispara donde se asume que estará el objetivo.
 - Considera que el robot enemigo se mueve a una velocidad constante en línea recta.
 - Efectivo cuando el objetivo está cerca.
 - Los movimientos de un robot pueden ser aproximados en una línea recta por un periodo muy corto de tiempo.
- ★ **Circular:** Dispara donde se asume que estará el objetivo.
 - Asume que el robot enemigo se mueve a una velocidad constante y a una velocidad angular constante.
- ★ **Oscilatorio:** Se asume que el robot enemigo está moviéndose hacia adelante y hacia atrás continuamente.
- ★ **Pattern Matching:** Dispara prediciendo una posición.
 - ✖ Guarda los movimientos pasados del robot y asume que éste se moverá siguiendo ese patrón.
 - ✖ Consume tiempo en el procesamiento de datos.

Ejecución de Robocode

Una batalla en Robocode repite los siguientes pasos (paso a paso en la batalla, ticks):

- 1°) La pantalla es reiniciada.
- 2°) Todos los robots ejecutan su código de manera simultánea hasta que se tomen las medidas.
- 3°) El tiempo es actualizado (tiempo = tiempo + 1)
- 4°) Todas las municiones se mueven y se chequean si hay colisión.

Jornadas de Capacitación de Capacitadores

Introducción

El profesorado del nivel medio, especialmente en las áreas de tecnología, computación, ofimática, deben actualizarse permanentemente para evitar las carencias en el desarrollo de las competencias digitales docentes, ya que día a día la brecha digital se incrementa entre docentes y alumnos. Si tenemos en cuenta que los adolescentes cada vez más se involucran con la tecnología, debemos aprovechar esta tendencia y generar espacios en la educación para aprovechar esas mentes y creatividad en pos de un futuro universitario.



Frente a esta realidad presentamos nuestra propuesta para alumnos secundarios: “Programa Incentivo de Nivel Medio sobre Lenguaje de Programación”, pero como expresamos en el párrafo anterior es necesario actualizar y renovar las competencias digitales de los docentes, razón por la cual incluimos en el proyecto las “Jornadas de Capacitación de Capacitadores”, cuyo propósito es promover y estimular en los docentes la potencialidad que la Programación posee para las prácticas educativas.

La idea es dirigirnos al personal docente del área informática computacional, para que adquieran competencias específicas sobre lenguaje de programación Java y software Robocode, con una metodología lúdica, cuyo incentivo es participar de una competencia entre colegios.

Marketing Educativo:

El marketing educativo es un proceso de investigación de necesidades sociales para desarrollar servicios educativos tendentes a satisfacerlas, acordes a un valor percibido, distribuidas en tiempo y lugar y, éticamente promocionadas para generar bienestar entre individuos y organizaciones.

A través de marketing educativo se invitará a los colegios secundarios de la ciudad de Córdoba a participar de las Olimpíadas Robocode que organiza el Instituto Universitario Aeronáutico (IUA), a su vez y como instancia previa necesaria se impulsarán las Jornadas de Capacitación de Capacitadores y el Programa Incentivo

de Programación para los estudiantes, una vez aprobado este último, los alumnos, se podrán inscribir en las Olimpiadas Robocode, representando a su colegio.

Aunque los más beneficiados son los alumnos, el IUA adquiere prestigio con este tipo de programa, además de ser una excelente herramienta de marketing para la institución.

El Instituto Universitario Aeronáutico otorgará Certificados a los participantes, tanto a los docentes como a los alumnos y al final de las Olimpiadas se entregarán premios a los ganadores. (Anexo 5, Certificados)

Respecto al **marketing educativo**, se utilizarán folletos, pancartas, videos y charlas explicativas, más propaganda on-line. Se hará difusión a través de la web y redes sociales y visitas a los colegios para entrega de propaganda gráfica.



Objetivos

Que el docente capacitador pueda:

- ▶▶ Analizar y valorar la importancia del empleo de estrategias y recursos lúdicos/didácticos como herramientas inherentes para un aprendizaje significativo.
- ▶▶ Adquirir competencias específicas en Lenguaje de Programación Java, y software Robocode, como entorno propicio para el desarrollo de propuestas de enseñanza de forma lúdica.
- ▶▶ Vincular el nivel medio con la Universidad, tendiendo a favorecer una mayor articulación institucional y pedagógica.

Temario

Durante estas Jornadas se actualizará y capacitará a los docentes sobre los temas específicos que incumben a nuestro “Programa Incentivo de Programación: Robocode” que consta de tres ejes principales:

- ▶▶ Lenguaje de Programación Java, como portador de información crucial de los conceptos básicos que nos introducen en esta área de conocimiento
- ▶▶ Software Robocode, como entorno propicio para el abordaje de esos contenidos y para favorecer a la construcción de aprendizajes
- ▶▶ La propuesta pedagógica, sobre la que se basa el Programa Incentivo de Programación, para desarrollar sus estrategias de enseñanza y plantear la forma de trabajo con los colegios secundarios.

Metodología

Las Jornadas de Capacitación de Capacitadores girarán en torno al Proyecto y estarán a cargo de profesores del IUA. Se dictará a todos los docentes de colegios secundarios interesados en este tipo de formación.

Nuestra propuesta es que se realicen durante cuatro días, un día por tema propuesto con una carga horaria de 6 hs., y un cuarto día para la evaluación final.

A medida que avancemos en las jornadas, se generarán espacios para la iniciativa de los participantes sobre el encare del temario planteado y aspectos técnicos a abordar, de forma de incentivar la realización de propuestas lúdicas y la actitud innovadora de estrategias educativas.

Para la obtención del certificado de aprobación del curso se requerirá al docente, el cumplimiento de los siguientes requisitos:

- ↘ 80% de asistencia
- ↘ Aprobación de la evaluación final

Planificación de las Jornadas



Día	Tema	Carga horaria	A cargo de
1	Lenguaje de Programación Java	6 hs reloj	Profesional del IUA
2	Software Robocode	6 hs. reloj	Profesional del IUA
3	Propuesta pedagógica	6 hs. reloj	Profesional del IUA
4	Evaluación Final	6 hs. reloj	Equipo examinador

Contenidos

Lenguaje de Programación Java



Los lenguajes de programación C y Fortran se han utilizado para diseñar algunos de los sistemas más complejos en lenguajes de programación estructurada, creciendo hasta formar complicados procedimientos.

No solo se necesitaba un lenguaje de programación para tratar esta complejidad, sino un nuevo estilo de programación. Este cambio de paradigma de la programación estructurada a la programación orientada a objetos, comenzó hace más de 30 años con un lenguaje llamado Simula67.

El lenguaje Java fue diseñado por James Gosling en SUN Microsystems. Desde los inicios de los años noventa hasta la actualidad se ha convertido en uno de los lenguajes de programación más utilizado, tanto para construir aplicaciones de escritorio multiplataforma, por aplicaciones web como incluso para aplicaciones para móviles usando Android.

Tres de las principales razones que llevaron a crear Java son:

1. Creciente necesidad de interfaces mucho más cómodas e intuitivas que los sistemas de ventanas que proliferaban hasta el momento.
2. Fiabilidad del código y facilidad de desarrollo.
3. Enorme diversidad de controladores electrónicos.

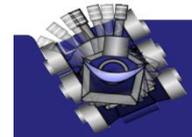
Partiendo de las nociones de variable, estructura condicional, estructura repetitiva y función, introduciremos al estudiante en la programación orientada a objetos, profundizando en el lenguaje Java.

Para instalar Java deberemos descargarnos un JDK desde la página oficial de Oracle (actual desarrollador) y continuar con las indicaciones del asistente de instalación.

Se puede consultar a la página oficial: <https://java.com/es/>

Página Oficial de Oracle:

<http://www.oracle.com/technetwork/es/java/javase/downloads/index.html>



Software Robocode

Robocode es un juego educativo de programación *Open Source* escrito en Java, cuyo objetivo es codificar un tanque de batalla robot para competir contra otros robots en un campo de batalla.

El jugador es el programador del robot, que no tendrá influencia directa en el juego. Pero, el jugador debe escribir la IA del robot diciéndole cómo comportarse y reaccionar sobre los acontecimientos que ocurren en el campo de batalla.

Para instalar Robocode se debe tener previamente instalado el JDK. Luego se debe descargar la última versión desde la página oficial, el archivo con "robocode-version.jar". En el caso de Windows se puede ejecutar directamente el archivo .jar, el cual va a descomprimir un directorio, y luego se ejecuta el archivo "robocode.bat" y sino en Linux se puede descomprimir y lanzar el ejecutable "robocode.sh"

Página oficial de Robocode: <http://robocode.sourceforge.net/>

Página API Robocode: <http://robocode.sourceforge.net/docs/robocode/>

Robocode Descarga: <http://sourceforge.net/projects/robocode/files/>

Propuesta pedagógica

En este tema se propone al docente, que motive a sus alumnos del área técnica computacional al conocimiento de los lenguajes de programación y software de juegos.

En la actualidad los adolescentes de la sociedad digital, se encuentran motivados con estas estrategias por considerarlas propias de su época. Una de las mayores ventajas del uso de los juegos en el aula es la capacidad de los estudiantes para involucrarse y comprometerse con el objetivo a ganar.

Los docentes deben tener en cuenta que los juegos ayudan a desarrollar un amplio espectro de habilidades en los alumnos, tales como: el pensamiento crítico, el razonamiento lógico, la memorización, la resolución de problemas y la toma de decisiones.

Además deben considerar que el taller es “intensivo”, razón por la cual pueden optar por contextos donde la participación de los alumnos pueda ser individual o formar parte de un grupo, en este último caso los estudiantes refuerzan la comunicación, habilidades de cooperación y negociación.

Por todo lo dicho, nos encontramos en un proceso de descubrimiento de nuevas estrategias y herramientas basadas en juegos y simulaciones para incorporarlas en el aula.

Entonces, la propuesta pedagógica que le sugerimos al docente es que debe:

- ▶ Organizar el contenido en forma lógica en secuencias de unidades como se propone en el Programa del Proyecto.
- ▶ Utilizar los juegos didácticos y las simulaciones para estimular la creatividad del alumno.
- ▶ Promover el autoaprendizaje a través de ensayo-error.
- ▶ Comprobar en forma directa y a través de ejercicios el avance de los resultados instructivos.

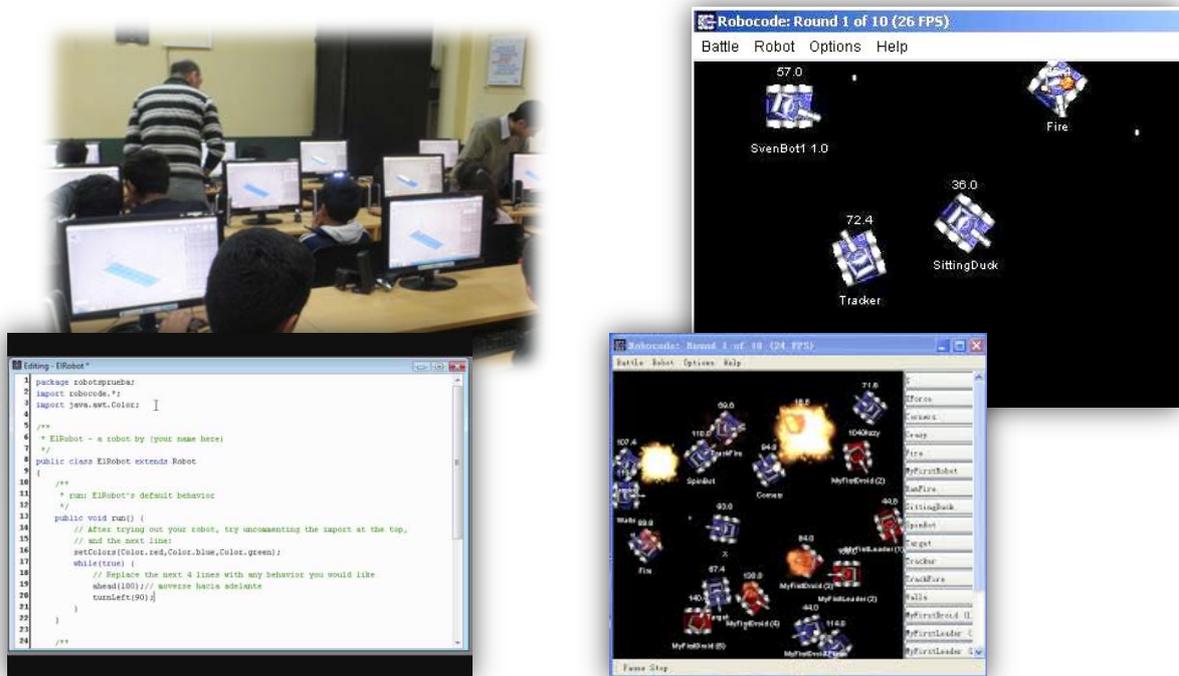
A modo de epílogo, y después de haber desarrollado los temas previstos, hacemos hincapié en el objetivo final de este Proyecto que es la participación de los colegios secundarios en las Olimpíadas de Robocode, considerando los objetivos, la estructura y los contenidos del taller propuesto.

Para ello los docentes deben evaluar a sus alumnos en el “Armado del Robot Final” el cual debe estar perfectamente capacitado para participar en dichas olimpíadas.

A la hora de evaluar a sus estudiantes

En cada colegio, al comienzo del Taller, se establecerá si el “trabajo final” será individual o en grupo.

El docente capacitador evaluará los robots y sus estrategias, además de rankear a los alumnos/equipos según el desarrollo optimizado de la estrategia de su robot.



Además, previamente, deberá leer la descripción realizada del robot desarrollado por el/los alumno/s, y que la misma sea acorde al nivel de la programación de dicho robot. Es decir, si la descripción, no desarrolla adecuadamente un nivel de estrategia avanzado (que haya sido implementado en el robot), se deberá solicitar una descripción más detallada.

Para poder puntuar los robots se desarrolló la **Guía de Puntuación**, que se da a continuación:

Condición / Concepto		Puntuación
Si existiese una copia de igual o más del 80% de un código conocido		Se considera inválido y se rechaza
Si el robot cuenta con los métodos run() y onScannedRobot() como mínimo		10 puntos
Evaluación del movimiento del robot	Movimiento simple	10 puntos
	Movimiento similar a robots dentro del código de ejemplo	15 puntos
	Movimiento basado en alguna estrategia avanzada, algoritmos explicados en la documentación de la Wiki de Robocode	30 puntos

Evaluación del movimiento de radar	Movimiento del radar de 360°, de manera arbitraria, que no afecte a su función de movimiento	10 puntos
	Movimiento fijo de radar a un robot en particular, ya sea implementando alguna estrategia (por energía, por nombre, por distancia, etc.)	20 puntos
	Movimiento de radar amplio, y no solamente a un robot en particular, sino que tenga amplio conocimiento del campo de batalla y elija el mejor blanco, según la estrategia implementada	30 puntos
Evaluación de la estrategia de disparo según la potencia	Energía de disparo constante	5 puntos
	Energía variable según la estrategia implementada	15 puntos
Evaluación de la estrategia de disparo según la estrategia	Disparo efectuado de manera normal con el método fire(), sin ningún tipo de estrategia	5 puntos
	Disparo implementando un cálculo predictivo lineal	30 puntos
	Disparo implementando un cálculo predictivo que se ajuste al movimiento del oponente	50 puntos
Por cada evento (definido en la API) desarrollado extra.		5 puntos
En el caso de declarar métodos propios agregando funcionalidad no provista por la API (Interfaz de programación de aplicaciones)		10 puntos

Los alumnos/equipos evaluados requieren obtener, por lo menos, 40 puntos para aprobar con la nota mínima.

Evaluación Final

Durante el cuarto día de jornada se llevará a cabo la evaluación final, la que se realizará a través de un cuestionario y ejercicios.

El Instituto Universitario Aeronáutico entregará Certificados a los docentes participantes. (Anexo 5 – Certificados Otorgados)

Síntesis

El docente se debe actualizar a la par del desarrollo tecnológico pues si no sabe aprender ¿cómo va a poder a enseñar a sus alumnos a aprender?

Las Jornadas de Capacitación de Capacitadores de nuestro Proyecto, instruirá a los docentes en la aplicación del lenguaje de Programación Java y el uso de la plataforma Robocode, como así también utilizar nuevas estrategias lúdicas en su proceso de enseñanza aprendizaje, para llegar a la instancia final que es la participación del colegio, a través de sus alumnos, en las Olimpíadas de Robocode en el IUA (competencia entre colegios secundarios de Córdoba).

Estos recursos pueden generar actividades de trabajo atractivas e innovadoras que sin su existencia sería imposible programar. Sin embargo, estos recursos por sí solos no pueden generar un cambio trascendental en la educación. Es el docente quien debe y puede originar ese cambio en las aulas auxiliado por esos recursos.

Utilizando el juego y el simulador, el docente tiene la posibilidad de achicar la brecha digital con el alumno y enseñarle a: analizar opciones, resolver problemas, tomar decisiones, realizar procedimientos, elaborar estrategias de combate, etc.

Con el fin de que su colegio juegue en las Olimpíadas de Robocode del IUA, el docente asistirá al/los alumnos desde el Taller de Programación hasta lograr el “robot final”, que será presentado para participar.

Para poder puntuar los robots y que sea un método uniforme para todos, desde nuestro proyecto, desarrollamos una Guía de Puntuación del Robot que el docente deberá utilizar.

Programa Incentivo Nivel Medio de Programación: Robocode

Introducción

El origen de este Proyecto es lograr la participación de adolescentes argentinos en las Olimpiadas de Robocode que ya se realizan a nivel mundial.

Para dar a conocer esta propuesta, realizaremos la debida difusión en los colegios a través de marketing educativo de forma impresa y digital.

Como expresamos anteriormente, para llevar a cabo este ambicioso Programa, necesitamos de dos eventos previos:

- ▶▶ Actualizar a los docentes de nivel medio sobre lenguaje de Programación Java y software Robocode.
- ▶▶ Lograr la motivación de los alumnos secundarios en aprender de forma lúdica Programación.

El primero ya lo detallamos en el título anterior y para el segundo, tomamos el juego Robocode como instrumento didáctico e innovador en el aula y, a alrededor de éste, generar múltiples actividades con docentes y alumnos.

El desafío de este Programa Incentivo que se desarrollará como **Taller**, se centra en la búsqueda de estrategias lúdicas que permitan el desarrollo e innovación de nuevas propuestas pedagógicas didácticas para promover la enseñanza de la Programación en el nivel medio de enseñanza.

¿Qué es un Taller?

Un taller es una modalidad de enseñanza y estudio caracterizada por la actividad, la investigación operativa, el descubrimiento científico y el trabajo en equipo que, en su aspecto externo, se distingue por el acopio, la sistematización y el uso de material especializado acorde con el tema para la elaboración de un producto tangible.

Robocode es un juego de Programación cuyo objetivo es programar la estrategia de un robot para competir contra otros robots en un campo de batalla. El jugador es el programador del robot y mediante el código que escribe le brindará inteligencia al robot, indicando cómo comportarse y reaccionar frente a eventos ocurridos en el juego.

Como dijimos anteriormente, al desarrollar nuestra investigación sobre “Acercar la Programación a las escuelas”, el espacio curricular apropiado para dictar el **Taller de Programación** es el 3° año del ciclo básico, donde los adolescentes están ávidos de estas nuevas estrategias lúdicas y herramientas didácticas para la enseñanza de Programación.

Objetivos

Que los alumnos de las escuelas secundarias:

- ▶▶ Aprendan el lenguaje de Programación, aplicando la creatividad y estrategias de combate.
- ▶▶ Sean capaces de programar robots en java.
- ▶▶ Apliquen la creatividad en pos de crear algoritmos que representen estrategias de combate del robot.
- ▶▶ Compitan en las Olimpiadas de Robocode.

Estructura del Taller

Cabe recordar que el Taller de Programación se realiza de modo presencial e “intensivo”, por lo que tendrá una duración de dos (2) días con una carga horaria de 8 hs. reloj cada uno.

Se dictará (en principio) en los colegios secundarios de la ciudad de Córdoba, que se hayan anotado para participar de las Olimpiadas Robocode, y donde sus profesores ya hayan realizado las Jornadas de Capacitación de Capacitadores, pues los docentes a cargo serán los mismos Capacitadores del colegio.



Los participantes del Taller serán los alumnos de 3° año (ciclo especialización), que voluntariamente se inscriban para participar de las Olimpiadas Robocode que se realizará en el Instituto Universitario Aeronáutico (IUA).

El Taller Incentivo de Programación para alumnos secundarios, está compuesto por dos módulos principales:

- Módulo I, corresponde a una introducción a la informática y al Lenguaje de Programación JAVA.
- Módulo II, corresponde a la Plataforma de Robocode, incluidas las reglas de juego.

Distribución de horas



1° Día de Taller:

Tema	Contenido	Carga horaria	Total hs
1	Presentación	30 min	8 hs.
2	Video Introductorio y Muestra de Robocode	30 min	
3	Java – Conceptos Básicos Parte I	2 hs.	
	Break	10 min	
4	Ejercicios de Java en Robocode	30 min	
5	Repaso de temas abordados	30 min	
6	Java – Conceptos Básicos Parte II	2 hs.	
	Break	20 min	
7	Java – Conceptos Avanzados	2 hs.	

2° Día de Taller:

Tema	Contenido	Carga horaria	Total hs
8	Ejercicios de Java en Robocode	1 hs.	8 hs.
9	Presentación de Robocode	30 min.	
10	Robocode – Parte I	1 hs.	
	Break	10 min	
11	Robocode – Parte II	1 hs.	
12	Ejercicios Prácticos Robocode – Ejercicios Básicos	2 hs.	
13	Ejercicios Prácticos Robocode – Ejercicios de Desafío	2 hs.	
	Break	20 min	
14	Ejercicios Prácticos Robocode – Armado de Robots Finales	30 min.	

Contenidos

Módulo Java

En primer término se dará una introducción a la informática, detallando qué es una computadora y cómo se organiza, para tener una idea general y una definición en común para comprender su funcionamiento. Luego se desarrollarán los lenguajes de programación, el cual hace referencia detenidamente paso a paso, cómo una computadora comprende las instrucciones dadas por un programador.

Tema	Java – Conceptos Básicos Parte I
1	¿Qué es una computadora?
2	Organización de una computadora
3	Lenguajes de Programación

Luego abordaremos Java propiamente dicho, las bibliotecas de clases y se dará un entorno típico de desarrollo.

Tema	Java – Conceptos Básicos Parte II
1	¿Qué es Java?
2	Biblioteca de clases
3	Entorno de desarrollo típico



Como final de este módulo se darán los conceptos avanzados de Java, donde se desarrollan definiciones de clases, variables, métodos, y demás aspectos técnicos básicos del lenguaje. Además se incluye el concepto de Programación Orientada a Objetos (POO).

Tema	Java – Conceptos Avanzados
1	Concepto de Programación Orientada a Objetos (POO)
2	Declaraciones de clases
3	Declaraciones de variables
4	Declaraciones de métodos
5	Operadores
6	Sentencias de decisión y bucles de repetición



Conocimientos mínimos de POO

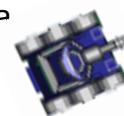
Los contenidos de la Programación Orientada a Objetos (POO) están diseñados para que el alumno pueda adquirir los conocimientos necesarios para:

- Comprender un modelo computacional del paradigma orientado a objetos.
- Comparar con otros paradigmas.
- Conocer los elementos y características de la POO.
- Comprender: Objetos, Instancias, Clases, Métodos, Variables de Instancias.
- Conocer que es Igualdad e identidad. Herencia. Superclase / Subclase.
- Entender el concepto de bytecode y de máquina virtual de Java.

Módulo Robocode

Como se mencionó anteriormente Robocode es un juego educativo *Open Source* escrito en *Java*, en el que los competidores deben diseñar y programar sistemas de control para unos tanques (robots), que compiten con el objetivo de eliminar el resto de los adversarios.

Para que el alumno se familiarice con este software, comenzamos con la presentación de Robocode y luego se abordarán los temas que se detallan a continuación, con los consiguientes conocimientos adquiridos para ellos:



Tema	Presentación de Robocode	Conocimientos Adquiridos
1	¿Qué es Robocode?	El alumno conocerá de qué se trata el juego, cuáles son sus componentes, cuál es el objetivo del mismo
2	¿Qué es Programación?	Comprenderá el concepto de Programación

Tema	Robocode Parte I – Reglas del Juego	Conocimientos Adquiridos
1	Propiedades iniciales de la partida.	<p>Conocerá las características que poseen los robots en tanto a energía, cómo se pierde ésta en cada disparo, cómo se regenera, la velocidad de disparo, el daño ocasionado por un disparo.</p> <p>Conocerá el sistema de puntuación de Robocode.</p> <p>Conocerá las estrategias de programación de acuerdo al estilo de lucha que pretenda el participante.</p>
2	Características generales de los robots (energía, disparo).	
3	Velocidad de la munición	
4	Daño realizado y daño recibido.	
5	Puntuaciones. Tipos de batallas (individuales o grupales).	
Robocode Parte I – Arena de Batalla		
1	Eje de coordenadas de la arena de batalla	Manejará el plano en que se mueven los robots. Esto es fundamental para guiar a los mismos y sirve como base para conocer las variables de movimientos del robot
2	Manejo de ángulos de la arena de batalla	Utilizará la orientación para movimientos angulares de los componentes de los robots

Tema	Robocode Parte II – Robots	Conocimientos Adquiridos
1	Partes de los robots.	<p>Conocerá qué elementos forman parte del robot, lo que es fundamental y base para su posterior programación.</p>
2	Concepto de variables	
3	Sensores de entorno de los robots (variables propias y variables obtenidas sobre el enemigo).	
Robocode Parte II – Comando de Robots		
1	Concepto de funciones.	Manejará y utilizará funciones que son muy importantes



Metodología

La metodología utilizada en el desarrollo del Taller incluye los siguientes tipos de actividades:

- ✧ **Actividad del profesor:** Clases expositivas sincronizadas con la realización de programas y ejercicios. Supervisión y asesoramiento en la resolución de los problemas por parte del alumno.
- ✧ **Actividad del alumno:** Toma de apuntes, participación activa en clase respondiendo a las cuestiones planteadas. Resolución de los ejercicios propuestos y escritura de pequeños programas durante el desarrollo de las clases.



Evaluación: Condiciones para la aprobación

- Se valorará la participación del alumno durante el Taller y se requerirá un trabajo final.
- El trabajo final será el “Armado de Robot Final”, donde los participantes, demuestren sus programaciones y estrategias trabajadas previamente, y donde se ponga en evidencia la comprensión y aplicación de los contenidos presentados en el Taller.
- El robot aprobado, será inscripto para las Olimpíadas de Robocode.

Síntesis

Los alumnos que tomen este Taller explorarán temas poco conocidos, en los que deberán desarrollar capacidades nuevas para pensar problemas mucho más complejos del nivel que solicita generalmente el colegio, y superarlos satisfactoriamente.

Así, pues lograrán conocimientos y desarrollos en programación básica Java, algoritmia, estructura de datos, Programación Orientada a Objetos y utilización de la plataforma Robocode como aspectos técnicos.

Al finalizar el taller, los participantes (alumno o equipo), serán capaces de realizar la programación de un robot, elaborar estrategias de combate y planificar la evaluación durante el proceso formativo.

Con el “robot final”, el/los alumnos podrán representar a su colegio en las Olimpiadas de Robocode que organiza cada año el Instituto Universitario Aeronáutico (IUA).

Antecedentes

Experiencia piloto Universitaria

En el Instituto Universitario Aeronáutico se realizó durante el año 2014, la primera Jornada del 1º Curso de Robocode a la que asistieron alumnos de 2º año de la Carrera de Ingeniería en Informática, de la Facultad de Ingeniería del IUA.

Se trató del primero de una serie de cursos dictado por el IUA (sólo plataforma Robocode, sin contenido de Java, para constatar si el nivel de uso de la plataforma era fácil de aprender) y en la Universidad Católica de Córdoba (UCC), como previa a un **Campeonato de Programación** que se llevó a cabo a finales del 2014, con sede en la institución.

En otra ocasión, se instruyó a los alumnos de colegios secundarios sobre lenguaje Java y la plataforma de Robocode. Fueron 2 jornadas de 8hs donde se hizo una introducción a la plataforma y unos ejercicios de práctica.



Referencias: <http://noticias.iua.edu.ar/spip.php?article360>
<http://www.institutodeanfunes.edu.ar/informatica/24/pagina.html>

Identificar nuestro Proyecto con los resultados obtenidos de la Experiencia piloto Universitaria en el Nivel Medio

Muchas facultades de ingeniería del país (para despertar vocaciones tempranas en los alumnos) presentaron planes y proyectos en las escuelas secundarias.

El Departamento de Informática y Computación de la Facultad de Ingeniería del IUA, presentó una propuesta para fortalecer los lazos entre la Universidad y el Nivel Medio.

Esta consistió en el dictado de un **Taller de Aprendizaje de Programación** de manera lúdica, en donde el estudiante secundario aplicó su ingenio y creatividad en el desarrollo de piezas de software, que representaban robots que competirían en lo que se ideó como etapa final: **“Las Olimpiadas de Programación Robocode IUA 2014”** (Ver Anexo 3 - Reglamento de las Olimpiadas Robocode 2014 en IUA), en las cuales, a modo de motivación para los alumnos participantes, se entregaron tres (3) premios.

La experiencia del IUA, se logró llevar a cabo en 3 (tres) escuelas del nivel medio:



La escuela ITeLF en la Falda. Participaron alumnos de 6to. Año. Esta escuela cuenta con una orientación técnica en electrónica, donde los alumnos, en una de sus materias, trabajaron con programación en C bajo nivel, por lo que contaban con una idea previa de programación, algoritmia, y razonamientos mínimos de programación.



La escuela IPEM N° 41. Se anotaron alumnos de 3er a 6to, Año. Es un colegio público de un barrio marginal de la Ciudad de Córdoba, en la cual los estudiantes nunca tuvieron trabajos relacionados a computación, por ende carecían de conocimiento alguno de cómo trabajaba una computadora, qué era un lenguaje de programación, para qué funcionaba y demás, los alumnos del IPEM N° 41 carecían de un centro de cómputos, por lo que cada alumno utilizaba su netbook del programa nacional “Conectar Igualdad” para el desarrollo de las actividades. Llegaron a las Olimpiadas los alumnos de 6to. Año.



La escuela Deán Funes. Competieron alumnos de 6to. Año. En la Ciudad de Córdoba, el Deán Funes se asemeja en parte al ITeLF, ya que los alumnos cuentan, entre las especializaciones la de Informática.

Es de suma importancia para nuestro Proyecto, el modo en que las autoridades de las instituciones se involucren en estos programas, tanto los directivos del nivel medio como los de las universidades, ya que ambos deben participar activamente en distintos aspectos.

- Por parte de las escuelas secundarias es sumamente importante motivar a los adolescentes a participar en estos tipos de taller, brindándoles las facilidades a las universidades para llegar a ellos.
- Por parte de las universidades poder dar soporte y apoyo tanto a docentes encargados del dictado de las clases, como así también para el desarrollo de las competencias interescolares.

Aspectos que caracterizaron a alumnos de las escuelas que tomaron la capacitación y realizaron la competencia

Cantidad de Alumnos que participaron por escuela	30	19	9
Conocimientos previos	Escuelas		
	ITeLF	IPEM N°41	Deán Funes
De la computadora	4	3	5
Concepto de programación	4	2	6
Conocimiento del lenguaje Java	2	1	2
Concepto de Objetos	2	1	2
Concepto de Clases	2	1	2
Concepto de Métodos	4	1	5
Concepto de Variables	4	1	5
Operadores	4	1	5
Sentencias de decisión y bucles de repetición	5	1	6
Algoritmos de Programación	4	1	6
Conocimiento de Robocode	1	1	2
Reglas del Juego	1	1	2
Arena de Batalla	1	1	1
Partes de un Robot	1	1	2
Comandos de Robots	1	1	1

Aspectos desarrollados por los alumnos en la capacitación y la competencia

De nuestra investigación, surge la siguiente estadística de alumnos que comprendieron y asimilaron los conocimientos generados en el Taller (lectura de la estrategia del robot) -> Sistema Experto, encuesta), para el desarrollo de sus robots y sus resultados en las olimpiadas.

Cantidad de Alumnos que participaron por escuela	30	19	9
Conocimientos adquiridos al final del Taller	Escuelas		
	ITeLF	IPEM N°41	Deán Funes
Conocimientos de una computadora	5	5	6
Concepto de programación	4	4	6
Conocimiento del lenguaje Java	5	5	5
Concepto de Objetos	5	5	5
Concepto de Clases	5	5	5
Concepto de Métodos	4	4	5
Concepto de Variables	4	4	5
Operadores	4	4	5
Sentencias de decisión y bucles de repetición	5	4	6
Algoritmos de Programación	4	4	6
Conocimiento de Robocode	5	5	5
Reglas del Juego	5	5	5
Arena de Batalla	5	5	5
Partes de un Robot	5	5	5
Comandos de Robots	5	5	5





Alumnos en el Taller de aprendizaje de Programación Robocode

Análisis de Resultados

Los resultados obtenidos por las experiencias citadas, fueron tentadores para presentar nuestro Proyecto de **Programa Incentivo Nivel Medio de Programación Robocode**, ya que la avidez de los adolescentes por aprender de manera lúdica, lenguajes de programación hacen interesante esta propuesta en los colegios secundarios.

A continuación presentamos nuestro análisis de resultados de la experiencia sobre las Olimpiadas realizadas en el IUA, que se adaptaría perfectamente a nuestro Proyecto.

En “**Las Olimpiadas de Programación Robocode IUA 2014**”, los alumnos de escuelas secundarias debían armar cada uno su propio robot, el cual era analizado antes de ser sumado en la competencia.

Debido a la cantidad de participantes inscriptos, las Olimpiadas se dividieron en dos etapas:

- 1°) Etapa inicial donde era una fase de grupos, las cuales se dividieron en 8 grupos de 5 participantes cada uno, y avanzaban los 2 mejores de cada grupo en una batalla de “rumble” (5 participantes al mismo tiempo).
- 2°) Segunda etapa, consistía en fases de eliminación directa, los participantes competían 1vs1 y el que ganaba pasaba a la siguiente ronda.

De las escuelas y alumnos participantes, resultó la siguiente estadística:

Cantidad de Alumnos que participaron por escuela	30	19	9
Descripción	Escuelas		
	ITeLF	IPEM N°41	Deán Funes
Alumnos que concurren al taller	19	9	30
Alumnos que presentaron su robot	17	3	20
% de presentaciones	89,47%	33,33%	66,67%

Donde se destaca:

- ➔ ITeLF, 7 participantes pasaron a la etapa de eliminación directa, 2 pasaron a cuartos, 1 a la semifinal quedando ese mismo en el 3er puesto.
- ➔ IPEM N° 41, de los 3 robots presentados pasaron la instancia inicial a la fase de eliminación directa, llegando uno de ellos a cuartos de final.
- ➔ Deán Funes, 7 participantes pasaron a rondas de eliminación directa, 5 pasaron a cuartos, 4 a semifinal, y luego completaron el 1er, 2do y 4to puesto.

Estas experiencias han permitido palpar una realidad de nuestra sociedad, muchos adolescentes están influidos por la tecnología y crecen día a día a la par de ella, está en nuestro interés captar esa necesidad y generar espacios para que puedan desarrollarla vinculando al estudiante secundario con el universitario, hacer del colegio un pasillo con la puerta abierta hacia la universidad.

Conclusiones

Las causas que dificultan la transición y continuidad de los alumnos secundarios a la universidad son variadas y múltiples: familia, situación socioeconómica, distancias, desconocimiento, etc. Ayudando a esta realidad, hay universidades que idearon planes y proyectos para despertar vocaciones tempranas en los estudiantes.

Basándose en nuestro proyecto, en la ciudad de Córdoba, Argentina, el Instituto Universitario Aeronáutico, realizó una experiencia piloto para acercar los colegios secundarios a la universidad.

Actualmente, en Argentina, la mayoría de las escuelas secundarias forman parte del plan nacional “Conectar igualdad”, constituyéndose las netbooks e Internet en un nuevo desafío áulico para el Área de Informática.

En este marco, planteamos la necesidad de articulación de contenidos y estrategias lúdicas de enseñanza entre ambos niveles educativos y las actualizaciones permanentes de las nuevas competencias digitales para docentes

La fascinación de los adolescentes por los juegos, videojuegos, Internet, etc., tiene un papel importantísimo para nuestro Proyecto, ese interés por la tecnología de los adolescentes debemos valorarlo y guiarlo hacia un futuro prometedor.

Consolidando el trabajo de actualización realizado con docentes, continuaremos con los estudiantes incorporando a sus conocimientos, lenguaje de programación Java y plataforma Robocode, estimulando la curiosidad, generando espacios exploratorios, colaborativos y experimentales.

Por todo lo expresado:

- ▶ Se propone un Programa Incentivo para Nivel Medio de Programación: Robocode, utilizando esta forma lúdica como incentivo para los adolescentes, el cual culmina con “Olimpiadas” donde demuestran su capacidad combatiendo entre ellos.
- ▶ Se presentan como instancia previa y necesaria las Jornadas de Capacitación de Capacitadores, las cuales actualizarán y renovarán las competencias digitales de los docentes secundarios.
- ▶ Se plantea acordar con las escuelas secundarias la introducción en su currícula, de un Taller de Programación cuyo contenido abarque el lenguaje Java y manejo de la plataforma Robocode.
- ▶ Se invita a los colegios secundarios a participar, anualmente, en las Olimpiadas de Robocode organizadas por el Instituto Universitario Aeronáutico (IUA)

Al realizar la investigación para este Proyecto, y considerando los resultados de la experiencia piloto del IUA, nos dimos cuenta que es totalmente viable que las Universidades se introduzcan en los Colegios Secundarios para despertar vocaciones tempranas a diferentes carreras y de concretarse permitirá influir cualitativamente en el sistema educativo, mejorando la educación técnica en ambos niveles educativos

Referencias / Bibliografías

Modelo pedagógico

- <https://gingermariatorres.wordpress.com/modelos-pedagogicos/>
- <http://www.joaquinparis.edu.co/DATA/MODELOS/PAGINAS/RAFAEL.htm>
- <http://www.monografias.com/trabajos26/modelos-pedagogicos/modelos-pedagogicos.shtml>
- http://es.wikipedia.org/wiki/Educaci%C3%B3n_tecnol%C3%B3gica
- http://bibliotecadigital.conevyt.org.mx/concurso/tematica_e/0132.pdf

Articulación Universidad-Escuela

- http://jets.linti.unlp.edu.ar/uploads/docs/presentacion_del_proyecto_actual.pdf

Marketing Educativo

- <http://conceptoaccionmarketing.blogspot.com.ar/2014/07/marketing-educativo.html>
- <http://mkteducacion.blogspot.com.ar/>
- https://es.wikipedia.org/wiki/Marketing_Educativo

TICs

- http://computingcareers.acm.org/?page_id=7

El juego como estrategia didáctica

- http://sedici.unlp.edu.ar/bitstream/handle/10915/41365/Documento_completo.pdf?sequence=1

Java

- http://es.wikipedia.org/wiki/Java_%28lenguaje_de_programaci%C3%B3n%29

Robocode

- Principal: <http://robocode.sourceforge.net/>
- API: <http://robocode.sourceforge.net/docs/robocode/>
- RoboWiki: http://robowiki.net/wiki/Main_Page
- Sitios de IBM:
 - <http://www.ibm.com/developerworks/java/library/j-robocode/>
 - <http://www.ibm.com/developerworks/java/library/j-robocode2/>
 - <http://www.ibm.com/developerworks/java/library/j-robotips/>
- <http://robocode.sourceforge.net/docs/robocode/>
- <http://robocoderepository.com>

Anexos

Anexo 1 – Programa Incentivo Nivel Medio de Programación: Robocode - Ejercicios

Anexo 2 – Códigos de Computación

Anexo 3 – Reglamento de las Olimpiadas Robocode 2014 en IUA

Anexo 4 – Ejemplo de Cronograma de Actividades

Anexo 5 – Certificados Otorgados

Anexo 6 – Formularios de Inscripción

Anexo 1 – Programa Incentivo Nivel Medio de Programación: Robocode - Ejercicios

Ejercicios de Movimiento

En esta serie de ejercicios se aprenderán movimientos con el robot, para ello, no se utilizarán técnicas de escaneo.



Movimiento 1: Implementar un robot que inicialmente avance de a 100 unidades. Luego, modificarlo de tal forma que avance indefinidamente.

★ Tip: Utilizar `Double.POSITIVE_INFINITY`.



Movimiento 2: Utilizando como base el robot anterior (Movimiento 1), implementar el código necesario para que avance hasta que choque con una pared. Posteriormente, mejorarlo para que retroceda utilizando la misma recta de movimiento.

★ Tip: Utilizar `onHitWall` event.



Movimiento 3: Utilizando como base el robot anterior (Movimiento 2), implementar el código necesario para que el robot se mueva de pared a pared en un ángulo ajustado múltiplo de 90. (Que avance y retroceda de forma vertical u horizontal).

★ Tip: Utilizar `getHeading() % 90`.



Movimiento 4: Crear un robot que se mueva por los límites del campo de batalla.

★ Tip: Similar a `Wall`.



Movimiento 5: Crear un robot que se desplace de forma circular, similar al robot `Spinbot`.

★ Tip: Similar a `Spinbot`.

Ejercicios de Radar

En esta serie de ejercicios se aprenderán movimientos del radar, para ello, no se utilizarán técnicas de movimiento, robot en estático.



Radar 1: Crear un robot que escanee en 360 grados. Posteriormente, modificar el robot con el fin de independizar el movimiento del radar del resto del robot.

★ Tip: Utilizar `setAdjustGunForRobotTurn`.



Radar 2: Crear un robot en el cual su radar siga a otro robot.

★ Tip: Similar a TrackFire.
Utilizar la fórmula `getHeading() getGunHeading() + e.getBearing()`.



Radar 3: (DESAFÍO) Crear un robot inteligente que pueda predecir la próxima posición de su objetivo y disparar.

★ Tip base: Leer e implementar la clase `EnemyBot`:
<http://mark.randomarticle.com/weber/java/ch4/lab3.html>
Leer e implementar la clase `AdvancedEnemyBot`:
<http://mark.randomarticle.com/weber/java/ch5/lab4.html>

★ Tip: Deberá crear 2 (dos) clases nuevas llamadas `EnemyBot` y `Advanced EnemyBot`, ver código de ayuda.

Ejercicios Conjunto

En esta serie de ejercicios se aprenderán movimientos de robot y radar, se deberán utilizar técnicas de movimiento y radar.



Conjunto 1: Implementar un robot que persiga a otro.

★ Tip: Similar a Ram.
Utilizar la librería `robocode.util.Utils.normalRelativeAngleDegrees`.
Utilizar para el giro la fórmula
`normalRelativeAngleDegrees(getHeading() + e.getBearing() getGunHeading())`.



Conjunto 2: (DESAFÍO) Utilizando como base el robot DustBunny, implementar un robot propio que pueda predecir la posición del enemigo antes de efectuar el disparo.

★ Tip: Ver páginas:

<http://old.robowiki.net/robowiki?DustBunny>

http://darkcanuck.net/rumble/robots/mld.DustBunny_3.8.jar

Anexo 2 – Códigos de Computación

Código de ayuda en los ejercicios:

```
import robocode.*;
public class EnemyBot {
    private double bearing;
    private double distance;
    private double energy;
    private double heading;
    private String name;
    private double velocity;

    public EnemyBot(){
        reset();
    }

    public void update(ScannedRobotEvent e){
        bearing = e.getBearing();
        distance = e.getDistance();
        energy = e.getEnergy();
        heading = e.getHeading();
        name = e.getName();
        velocity = e.getVelocity();
    }

    public void reset(){
        bearing = 0;
        distance = 0;
        energy = 0;
        heading = 0;
        name = "";
        velocity = 0;
    }

    public double getBearing(){
        return bearing;
    }

    public double getDistance(){
        return distance;
    }

    public double getEnergy(){
        return energy;
    }

    public double getHeading(){
        return heading;
    }
}
```

```

public String getName(){
    return name;
}

public double getVelocity(){
    return velocity;
}

public boolean none(){
    if(name.equals("")){
        return true;
    }else{
        return false;
    }
}
}

import robocode.*;
public class AdvancedEnemyBot extends EnemyBot {
    private double x;
    private double y;

    public double getX(){
        return x;
    }

    public double getY(){
        return y;
    }

    public void reset(){
        super.reset();
        x = 0;
        y = 0;
    }

    public AdvancedEnemyBot(){
        reset();
    }

    public void update(ScannedRobotEvent e, Robot robot){
        super.update(e);
        double absBearingDeg = (robot.getHeading() + e.getBearing());
        if(absBearingDeg < 0) absBearingDeg += 360;
        x = robot.getX() + Math.sin(Math.toRadians(absBearingDeg)) *
e.getDistance();
        y = robot.getY() + Math.cos(Math.toRadians(absBearingDeg)) *
e.getDistance();
    }
}

```

```
public double getFutureX(long when){
    return x + Math.sin(Math.toRadians(getHeading())) * getVelocity() *
when;
}

public double getFutureY(long when){
    return y + Math.cos(Math.toRadians(getHeading())) * getVelocity() *
when;
}
}
```

Anexo 3 – Reglamento de las Olimpiadas Robocode

Olimpiadas de Programación Robocode IUA 2014

Bases

Agenda

Envío de Robots	Hasta el 29 de Octubre	Por mail (ver Participación/punto 7)
Eliminatorias y Final	5 de Noviembre	Laboratorio A y Java Lab / Facultad de Ingeniería del IUA

Participación

1. Se podrán realizar equipos con 2 (dos) participantes como máximo.
2. Una vez enviado el robot no se permite modificación alguna de su código.
3. Los equipos programarán su robot utilizando sólo las clases existentes en la API de Robocode.
4. El código desarrollado y la lógica del robot deberá ser debidamente argumentados por los participantes ante un jurado el día del torneo con el fin de comprobar su autoría.
5. Cada equipo, antes de iniciar la competencia, expondrá ante los jurados el funcionamiento de su robot y el porqué de su estrategia. Esta exposición no debe superar los tres minutos.
6. Si se descubre que un equipo ha realizado copia de otro robot serán inmediatamente descalificados sin tener en cuenta su posición dentro del torneo.
7. Deberá inscribirse enviando un mail a magm@iua.edu.ar con copia a nicolasquini@gmail.com, adjuntando el código fuente del robot y consignando en el cuerpo del mail exactamente los siguientes datos:
 - a. Nombre y Apellido:
 - b. DNI:
 - c. Universidad (si corresponde):
 - i. Facultad:
 - ii. Carrera:
 - iii. Curso:

- d. Colegio (si corresponde):
 - i. Año y Curso
- e. Email de contacto:
- f. Nombre (Alias) del robot:
- g. Explicación de la estrategia del Robot en no más de 200 palabras. Los puntos a. hasta f. deberán repetirse en caso de ser un equipo de dos participantes, el punto g. será único por equipo.

Competencia

1. El torneo de Robocode estará basado en la versión de Robocode 1.9.2.1.
2. Las batallas se llevarán a cabo entre tanques diseñados por grupos de dos personas en batallas (1 vs 1).
3. El robot que logre destruir al otro ganará la ronda inmediatamente, en caso de que no se destruyan se revisará en la tabla de Ranking de Robocode el robot que tenga el mejor puntaje y ese ganará la ronda.

Reglas del Torneo

1. El tamaño del campo de batalla (Battlefield Size) será de 800x600.
2. Los arcos de escaneo (Scan Arcs), la energía (Robot Energy) y los nombres de cada robot (Robot Name) serán siempre visibles.
3. Los cuadros por segundo serán fijados a 30fps.
4. La tasa de enfriamiento del cañón (Gun Cooling Rate) será de 0.1.
5. El tiempo de inactividad (Inactivity Time) será de 450.
6. Si se produce un estancamiento, empate de puntajes entre los robots o algún problema de software durante algún combate, se reiniciará la partida.
7. Los robots serán compilados con el sistema ROBOCODE instalado en el equipo sobre el cual se llevará a cabo el torneo. Si un robot no compila por algún motivo, se dará la oportunidad al participante para corregir el problema, el tiempo de corrección no debe superar los cinco minutos, pasado ese tiempo, el robot no será parte del campeonato.

Premios

- 1°) Primer Puesto: Consola de Videojuegos Xbox 360.
- 2°) Segundo Puesto: Tablet
- 3°) Tercer Puesto: Tablet

Otras Consideraciones

1. El jurado se reserva la posibilidad de eliminar un robot si detecta fraude.
2. El jurado se reserva el derecho de modificar las bases del concurso.

Anexo 4 - Ejemplo de Cronograma de Actividades

Proyecto Programa Incentivo de Nivel Medio de Programación: Robocode

Actividad	Abril	Mayo	Jun	Agos	Set.	Oct.	Nov.
Difusión en las escuelas	X	X					
Inscripciones			X				
Jornadas de Capacitación de Capacitadores – IUA (*)				X			
Taller de Programación (**)					X	X	
Olimpiadas Robocode – IUA (***)							X

(*) A las Jornadas de Capacitación asistirán los docentes de todos los colegios que se inscriban para las Olimpiadas Robocode.

(**) Los Talleres de Programación serán dictados, en cada colegio, por los Capacitadores que hayan realizado las Jornadas.

(***) Quedarán inscriptos para las Olimpiadas aquellos alumnos/equipos cuyo “robot final” sea aprobado según la Guía de Puntuación establecida en este Proyecto.

Anexo 5 – Certificados Otorgados

Ejemplos de Certificados entregados por el IUA





INSTITUTO UNIVERSITARIO AERONÁUTICO

Certificado de Asistencia

Por la presente certifico que (Apellido Nombre), DNI (.....),
ha obtenido en la Olimpiada de Robocode:

“Cuartos de Final”

Lic. Alicia Salamon
Dir. Departamento Informática

Ing. Alejandra Boggio
Dir. Ingeniería Informática

Ing. Mariano Garcia Mattio
Coordinador



INSTITUTO UNIVERSITARIO AERONÁUTICO

Certificado de Asistencia

Por la presente certifico que (Apellido Nombre), DNI (.....),
ha obtenido en la Olimpiada de Robocode:

“Mejor Estrategia”

Lic. Alicia Salamon
Dir. Departamento Informática

Ing. Alejandra Boggio
Dir. Ingeniería Informática

Ing. Mariano Garcia Mattio
Coordinador



INSTITUTO UNIVERSITARIO AERONÁUTICO

Certificado de Asistencia

Por la presente certifico que (Apellido Nombre), DNI (.....),
ha obtenido en la Olimpiada de Robocode:

“1er Puesto”

Lic. Alicia Salamon
Dir. Departamento Informática

Ing. Alejandra Boggio
Dir. Ingeniería Informática

Ing. Mariano Garcia Mattio
Coordinador

Anexo 6 – Formularios de Inscripción

Formulario de Inscripción del Colegio a las Olimpíadas Robocode



OLIMPIADAS ROBOCODE INTERCOLEGIALES

FORMULARIO DE INSCRIPCIÓN

Invitamos a los Colegios del nivel medio a participar de estas Olimpíadas Intercolegiales.

El objetivo es programar un robot para competir en un campo de batalla.

*Como instancias previas y necesarias a la inscripción del Colegio, deberán haber realizado las **Jornadas de Capacitación de Capacitadores** y el **Taller de Programación** para los alumnos.*

Deberán inscribirse enviando un mail a magm@iua.edu.ar con copia a nicolasguini@gmail.com, adjuntando el código fuente del robot y consignando en el cuerpo del mail exactamente los siguientes datos:

- a) *Nombre y Apellido:*
- b) *DNI:*
- c) *Colegio:*
- d) *Año y Curso:*
- e) *Email de contacto:*
- f) *Nombre (Alias) del robot:*
- g) *Explicación de la estrategia del Robot en no más de 200 palabras.*

Los puntos a) hasta f) deberán repetirse en caso de ser un equipo de dos participantes, el punto g) será único por equipo.

Premios:

Primer Puesto: Consola de Videojuegos Xbox 360.

Segundo Puesto: Tablet

Tercer Puesto: Tablet

Para más información consultar el Reglamento Olimpíadas Robocode.



JORNADAS DE CAPACITACIÓN DE CAPACITADORES

Invitamos a los docentes del Área Técnica Computacional a las Jornadas de Actualización en Lenguajes de Programación

El objetivo es desarrollar instancias de actualización y capacitación a los docentes, para instruir a sus alumnos con propuestas de enseñanza de forma lúdica.

La meta es que los alumnos lleguen a la instancia de poder participar en las Olimpíadas de Robocode.

Temario: Lenguaje JAVA, Software ROBOCODE, Estrategias pedagógicas

Días

Horario:

Inscripciones: