

Introducción a la representación del conocimiento y el razonamiento en educación pre-universitaria mediante una actividad STEM

Sara Guerreiro-Santalla, Oscar Fontenla-Romero, Alejandro Romero, Francisco Bellas

CITIC Research Center
Universidade da Coruña
A Coruña

sara.guerreiro@udc.es, oscar.fontenla@udc.es, alejandro.romero@udc.es,

francisco.bellas@udc.es

Resumen

Este trabajo presenta un recurso docente de acceso libre, orientado a estudiantes de últimos años de secundaria o bachillerato, centrado en el ámbito de la enseñanza de la Inteligencia Artificial (IA) pre-universitaria. En concreto, se plantea una actividad para introducir los conceptos de representación del conocimiento y razonamiento en IA con un enfoque STEM. En esta actividad, los estudiantes desarrollan completamente una aplicación para *Smartphone* mediante el entorno de programación App Inventor, siguiendo una metodología de aprendizaje basada en proyectos. Exactamente, la aplicación a desarrollar se denomina "The School Path Guide", y tiene como objetivo guiar a los visitantes de un centro educativo de un punto a otro por la ruta más corta utilizando imágenes en primera persona, y considerando posibles situaciones de colapso. Los estudiantes deben realizar un mapa en planta de su centro, tomar fotos de las posibles rutas, crear a partir del mapa una representación en forma de grafo, y añadir a éste probabilidades empíricas de colapso en los enlaces. La programación debe representar adecuadamente el grafo, y debe permitir un razonamiento probabilístico sencillo sobre el mismo. El recurso planteado se ha validado en el marco del proyecto Erasmus+ denominado AI+, liderado por la Universidade da Coruña, con resultados muy satisfactorios.

Abstract

This paper presents an open-access teaching resource, aimed at students in their final years of secondary school or high school, focused on the field of pre-university Artificial Intelligence (AI) education. Specifically, an activity is proposed to introduce the concepts of knowledge representation and reasoning in AI with a STEM approach. Students fully develop a

Smartphone application using the App Inventor programming environment, following a project-based methodology. Specifically, the developed application is called "The School Path Guide", and it aims to guide visitors of a school from one point to another by the shortest route using first-person images and considering possible collapse situations. Students must make a floor plan map of their school, take pictures of possible routes, create a graph representation from the map, and add empirical probabilities of collapse at the links. The programming must adequately represent the graph and it must allow for simple probabilistic reasoning about it. The proposed resource has been validated within the framework of the Erasmus+ project called AI+, led by the University of A Coruña, with very satisfactory results.

Palabras clave

Educación en inteligencia artificial, STEM, aprendizaje basado en proyectos, razonamiento probabilístico, representación mediante grafos.

1. Introducción

El Plan Europeo de Educación Digital 2021-2027 [5] establece, como una de las prioridades para los estados miembros, el desarrollo de currículos (planes de estudio) formales para la enseñanza de la Inteligencia Artificial (IA) en la etapa pre-universitaria. Esta recomendación sigue la estela del gran avance en este ámbito que ya existe en otros países como EEUU [9], China [2] o India [3]. Mientras los responsables educativos están desarrollando planes globales, se potencia la creación de material educativo de calidad que pueda ser utilizado por la comunidad educativa.

Es en este contexto donde se encuadra el recurso docente que aquí se presenta. Ha sido implementado y probado en el ámbito del proyecto AI+ [1], una inicia-

tiva de innovación educativa que ha sido financiada por la Unión Europea a través del programa Erasmus+ en 2019. Engloba a seis centros de enseñanza secundaria de cinco países diferentes (España, Italia, Eslovenia, Lituania y Finlandia), con el objetivo de desarrollar un currículo introductorio de IA para estudiantes de entre 15 y 17 años. El proyecto está liderado por la Universidade da Coruña (UDC), que implementa las unidades didácticas (UD) específicas que componen el plan de estudios. Contienen conceptos teóricos sobre los 8 temas de IA previamente definidos (Percepción, Actuación, Representación, Aprendizaje, Razonamiento, Motivación, IA colectiva y Ética), así como actividades específicas para trabajarlos en el aula [7]. Estas UD son revisadas por los profesores de los seis centros y probadas con sus alumnos, lo que proporciona retroalimentación para mejorarlas. Finalmente, las UD se publican en la web del proyecto para ponerlas a disposición de toda la comunidad docente [1]. Hay que destacar que, recientemente, el trabajo realizado en este proyecto ha sido la base de una nueva asignatura oficial de bachillerato (Introducción a la IA), ofertada a todos los centros de secundaria de la comunidad autónoma de Galicia en el curso 2021/22 [17].

El enfoque de IA que se sigue en AI+ es el de *agente inteligente*. Los sistemas de IA se presentan a los alumnos como agentes que reciben percepciones del entorno, real o simulado, y realizan acciones para cumplir sus objetivos de forma autónoma [14]. Esta perspectiva de la IA está en consonancia con las recomendaciones de la Comisión Europea [4] para los futuros planes de educación digital, centrados en la "IA específica", es decir, la IA que se implementa en sistemas reales que resuelven problemas específicos a las personas.

Para poder implementar este enfoque en las aulas es necesario contar con elementos hardware específicos. Para ello, dentro de AI+ se ha establecido utilizar el *Smartphone* del alumno como elemento tecnológico básico para todo el material didáctico a desarrollar. Los *Smartphone* actuales tienen el nivel tecnológico necesario para la enseñanza de la IA en cuanto a sensores, actuadores, potencia de cálculo y comunicaciones, y lo tendrán también en el futuro dada su continua actualización. Además, la gran mayoría de los estudiantes de bachillerato tienen su propio *Smartphone*, que reduce significativamente el coste de la introducción de esta disciplina en las aulas e iguala regiones con diferente capacidad económica.

Para introducir los 8 temas de IA comentados anteriormente utilizando el *Smartphone*, los alumnos se enfrentarán a problemas reales pertenecientes a tres campos de aplicación en las UD: *apps* inteligentes para *Smartphone*, robótica autónoma y entornos inteligentes (Internet de las Cosas). Aunque se podrían haber seleccionado muchos otros campos, estos son

muy representativos del dominio actual de la IA, y todos ellos pueden ser desarrollados en los centros educativos utilizando un *Smartphone* mediante App Inventor [11], el robot Robobo [15] y el sistema Home Assistant [8].

Antes de presentar los detalles del recurso concreto que aquí se describe, hay que señalar que el currículo de AI+ sigue un enfoque *STEM*, ya que requiere conocimientos de diferentes disciplinas para resolver los casos prácticos, como *física, matemáticas o programación*. Además, cada UD presenta un reto o proyecto que debe ser afrontado a través de un enfoque de aprendizaje cooperativo basado en proyectos (CPBL) [10], donde los estudiantes se organizan en grupos y realizan los pasos típicos de un proyecto de *ingeniería*. Esta metodología se basa en un aprendizaje proactivo en el que se refuerzan los conceptos teóricos a medida que se les exige que resuelvan una cuestión práctica (*learning by doing*). Por último, hay que aclarar que las unidades didácticas que componen el currículo de AI+ están diseñadas *para el profesor*, por lo que no deben ser proporcionadas directamente a los alumnos, sino que se utilizan para que el profesor las adapte a sus grupos de forma adecuada.

2. Descripción del recurso

Este recurso se corresponde con el reto que los alumnos deben afrontar en la tercera unidad didáctica del proyecto AI+, dedicada a la introducción a los conceptos de *representación del conocimiento y razonamiento en IA*. En esta actividad, los alumnos desarrollarán una aplicación inteligente para el *Smartphone* utilizando el software MIT App Inventor. Esta *app* ha sido bautizada como "School Path Guide", ya que guía a los usuarios del edificio escolar (visitante, profesor, alumno...) desde su ubicación actual hasta un destino que ellos seleccionen, basándose en fotografías, texto e indicaciones de voz. Se propone utilizar una representación de la escuela basada en fotografías de lugares representativos que el usuario pueda identificar fácilmente. Para formalizar dicha representación, se introduce a los alumnos el concepto de mapa topológico, y tienen que crear una representación tipo grafo que posteriormente se utilizará para realizar razonamientos probabilísticos básicos en función de la información en tiempo real.

Todo el material didáctico relacionado con la actividad está disponible para su descarga [13]. El archivo "TU3_Representation_Reasoning.pdf" contiene la descripción de la actividad en sí (en inglés). En los siguientes sub-apartados se ofrecen más detalles sobre el recurso y su ámbito de aplicación.

2.1. Edad recomendada y pre-requisitos

La actividad ha sido diseñada para estudiantes de bachillerato. Para que puedan alcanzar adecuadamente

los objetivos de aprendizaje del recurso, deben tener los siguientes conocimientos previos:

- Programación: nivel básico sobre programación basada en bloques, condicionales, bucles, variables, listas y funciones.
- Conocimientos básicos de MIT App Inventor.
- Matemáticas: representación 2D de mapas, planos y coordenadas cartesianas.
- Conocimientos previos recomendados: es recomendable que los alumnos tengan algunos conocimientos básicos sobre cómo se almacenan y representan los datos en un ordenador.

2.2. Organización temporal y material

Dividir un reto global en tareas que lleven a su realización es una competencia clave en la metodología CPBL que los alumnos deben adquirir. En esta actividad, dicha división se proporciona a los profesores para que puedan guiar a todos los estudiantes de forma homogénea, si así lo desean. En este caso se contemplan dos actividades, y cada una de ellas implica dos tareas diferentes. La duración global de la actividad es de 8 horas, aproximadamente.

En la unidad se propone que la realización de esta actividad se realice en grupos de dos alumnos, de cara a maximizar su implicación en las tareas. Cada uno de ellos tendrá un rol: *programador* o *gestor*. Aunque en este caso el objetivo principal es que ambos alumnos colaboren en la programación de la *app*, el programador se centrará en el desarrollo del programa en App Inventor exclusivamente. Por otro lado, el gestor deberá encargarse de todos los aspectos necesarios para llevar a cabo la *app* de forma adecuada, como tomar notas, hacer preguntas, gestionar el tiempo y presentar los entregables al profesor. Es muy importante que ambos miembros estén siempre de acuerdo y sean conscientes de lo que hace el otro.

Los recursos necesarios para seguir la unidad didáctica son los siguientes:

1. Un portátil u ordenador por grupo conectado a una red WI-FI.
2. Un Smartphone Android por grupo, conectado a la misma WI-FI, y con dos apps instaladas: App Inventor y un lector de códigos QR.
3. Cada grupo debe tener una cuenta activa de App Inventor.
4. Plantilla de la app: para reducir la duración de la actividad, se proporciona un proyecto de App Inventor [13] como plantilla para centrar la atención en los aspectos de IA y no en los de programación. Es el profesor quien decide si los alumnos deben realizar la enseñanza de forma abierta, o si prefiere que sigan la plantilla. El archivo para los estudiantes, una posible solución y una des-

cripción de los bloques que contiene están disponibles para su consulta [13].

5. Códigos QR: se propone crear los QR que se utilizarán posteriormente en un software o web adecuados, e imprimirlos (tamaño recomendado 10x10cm).

2.3. Conceptos de IA trabajados

El recurso aquí presentado ofrece una introducción a la representación y el razonamiento en el ámbito de la IA [6]. Se supone que ambos temas son nuevos para los alumnos de esta edad, por lo que esta actividad pretende presentar sus fundamentos con una aplicación real sencilla pero ilustrativa. Desde un punto de vista teórico, el profesor debe explicar previamente los fundamentos de la representación en este contexto, explicando las bases de la representación digital de la información que usan los ordenadores sobre los cuales se ejecutan los sistemas de IA. También se deben introducir los enfoques de representación simbólica y sub-simbólica de manera sencilla. Por otro lado, se debe proporcionar una introducción básica al razonamiento en IA [12], relacionándolo con la resolución automática de problemas, y explicando de manera básica conceptos como los algoritmos de búsqueda o los sistemas basados en reglas. Lo más importante de esta parte teórica es dejar clara la *relación existente entre representación y razonamiento*, y cómo la primera condiciona en gran medida a la segunda [16].

Desde un punto de vista práctico, en esta actividad se tratarán los siguientes conceptos específicos de representación y razonamiento en IA: mapas topológicos vs métricos, representación en primera persona, definición básica de grafos, razonamiento probabilístico, y búsqueda básica de rutas sobre grafos.

2.4. Evaluación

Para aprovechar al máximo este recurso, se propone un esquema de evaluación específico, adaptado al enfoque metodológico propuesto y que ha sido contrastado dentro del proyecto AI+:

- *Prueba final del programa*: cada grupo debe mostrar su *app* en funcionamiento normal, es decir, con un visitante que se mueve por el centro educativo. El profesor debe realizar una comprobación de esta solución final, revisando si se han tenido en cuenta todas las especificaciones, y si se puede utilizar con fiabilidad, principalmente, si la respuesta es estable en diferentes ejecuciones. Además, los alumnos deben entregar su código de App Inventor, para que el profesor pueda probarlo si es necesario. Como no se trata de una actividad de programación, el énfasis de la evaluación no debe estar en la calidad del código, sino en su funcionamiento.

- *Prueba final de conceptos teóricos*: al final de la actividad, los alumnos deberán rellenar un cuestionario centrado en los principales conceptos de IA tratados, representación y razonamiento. Se ha incluido un ejemplo concreto de cuestionario en el material disponible [13].
- *Trabajo diario*: la implicación de los alumnos durante el desarrollo de la actividad debe ser evaluada, siendo un aspecto muy importante en la evaluación de cualquier metodología CPBL. Esta evaluación debe considerar el trabajo desarrollado en cada uno de los roles asignados. La UD contiene rúbricas individuales que el profesor debe rellenar cada semana [13].

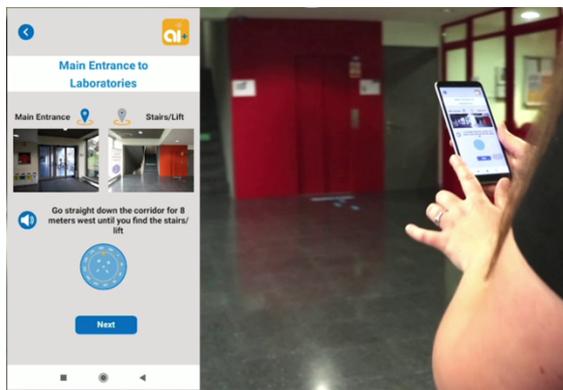


Figura 1: Interfaz y uso de la app desarrollada.

3. Actividades y tareas

Desarrollar una versión funcional de la app “The School Path Guide” constituye el objetivo final del proyecto que los estudiantes deben llevar a cabo en este recurso educativo. Según la metodología CPBL, sus especificaciones deben quedar claramente establecidas al inicio de la clase. Debe ser una app apta para cualquier tipo de centro educativo, y útil para aquellos que no conocen el centro, es decir, los visitantes. Debe funcionar de la siguiente manera: existen diferentes puntos de localización en distintos lugares del edificio (entrada principal, aulas, biblioteca...) identificados por un código QR, que ha sido codificado con el nombre de la localización. Cuando el usuario llega a uno de estos puntos, escanea el código QR a través de la app utilizando la cámara del Smartphone. Una vez escaneado, la aplicación muestra una lista de posibles destinos y el usuario selecciona el deseado. A partir de ese momento, la app muestra el camino óptimo a seguir para llegar al destino a través de fotos e instrucciones que aparecen en la pantalla. El funcionamiento básico de la app se muestra en el vídeo llamado "SchoolPathGuide.mov" (ver captura en Figura 1), incluido en el material de la unidad [13]. Los alumnos deben verlo al comienzo de la actividad para entender el tipo de solución que tienen que conseguir.

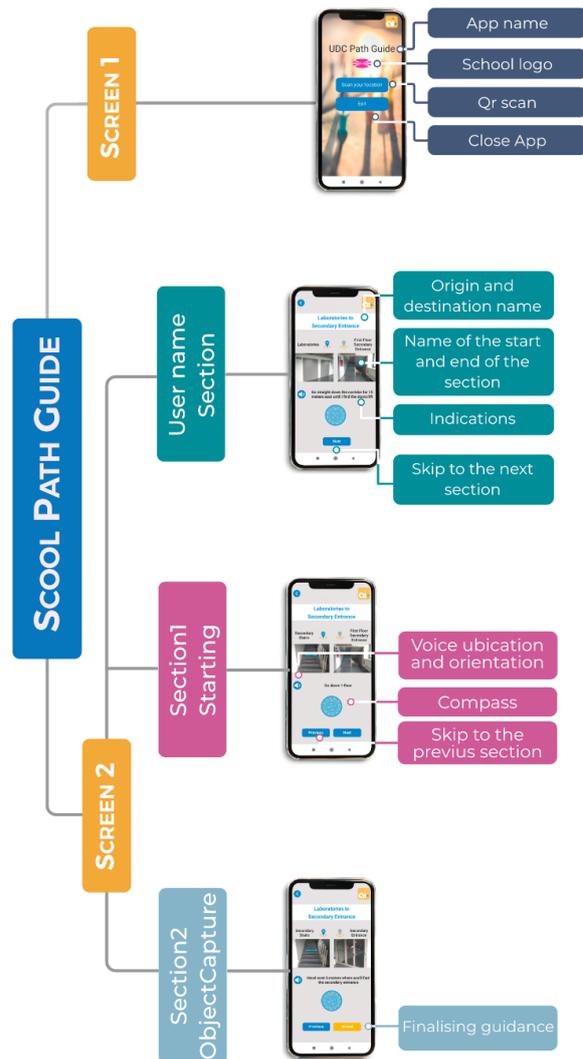


Figura 2: Esquema del contenido de la plantilla.

La plantilla de App Inventor, disponible también como recurso [13], contiene un conjunto de bloques pre-programados que proporcionan un soporte básico al funcionamiento anterior. La estructura funcional de dicha plantilla se puede observar en el esquema de la Figura 2, que también está disponible para los profesores. El interfaz de usuario se estructura en dos pantallas diferentes: *Pantalla1* donde se encuentra el botón de escaneo del QR, y *Pantalla2* donde el usuario puede seleccionar el destino y donde, una vez seleccionado, se muestra el camino a seguir mediante imágenes e indicaciones de texto. Además, la *Pantalla2* contiene una representación del sensor de orientación, y otros botones para avanzar o volver al paso anterior de la ruta. Se pueden observar capturas de estas pantallas en la Figura 2.

Obviamente, es posible que el profesor considere que los alumnos deben realizar toda la programación sin utilizar esta plantilla, dando libertad para la creación de las pantallas. De todos modos, según la meto-

dología del CPBL, se deben seguir respetando las especificaciones de funcionamiento de la aplicación establecidas en el documento [13].

3.1. Actividad 1: representación del conocimiento

El primer paso para desarrollar la *app* es definir la representación del centro educativo que se utilizará internamente en la aplicación, para luego poder encontrar la ruta óptima basada en ella. Uno de los tipos más sencillos de representación de un edificio es el *plano*, que proporciona una vista en planta que suele abarcar una zona amplia. Realizar un guiado útil de la trayectoria a través de una representación en planta implica conocer la ubicación del usuario para mostrar su posición mientras se mueve, pero en interiores no es fácil (en exteriores podríamos utilizar el GPS). Por ello, como primera aproximación, se propone utilizar otra representación de la escuela basada en imágenes, es decir, fotografías de lugares que el usuario pueda identificar fácilmente (*puntos de localización*). Se trata de una visión en "primera persona" de la escuela que abarca un campo de visión más reducido, pero que permite mostrar el camino a seguir de forma sencilla, simplemente creando una secuencia ordenada de pasos. Así, la localización del usuario no es tan relevante porque puede conocer su posición viendo las imágenes y comparándolas con el mundo real.

Los *puntos de localización* conforman una representación *topológica* simple del mapa del centro. Para entender este tipo de representación, los profesores deben revisar, y luego explicar a los alumnos, las diferencias entre los mapas topológicos y los métricos:

- La cartografía métrica es la más habitual para los alumnos y considera un espacio bidimensional en el que se sitúan los objetos con coordenadas precisas, por ejemplo, coordenadas cartesianas.
- La cartografía topológica sólo considera los lugares relevantes y las relaciones entre ellos. A menudo, también se almacenan las distancias entre los lugares. El mapa es un grafo, donde los nodos corresponden a lugares y los arcos a recorridos.

El primer paso para los alumnos es definir estos *puntos de localización*, que se marcarán con un código QR colocado en una pared en una posición visible. Estos *puntos de localización* conforman los posibles orígenes y destinos de los usuarios en la escuela. Aparecerán listados en la aplicación y es muy importante seleccionarlos adecuadamente. Se recomienda, para la primera versión de la *app*, definir un máximo de 4 puntos distribuidos en dos plantas (baja y primera). Con este conjunto puntos de localización, el número máximo de rutas posibles es de 12, correspondiendo a cada punto de localización hacia 3 posibles destinos. Se podrían definir 12 recorridos diferentes y crear 12 secuencias de imágenes distintas para

representarlos, pero hay muchos tramos que son coincidentes, principalmente en el edificio escolar.

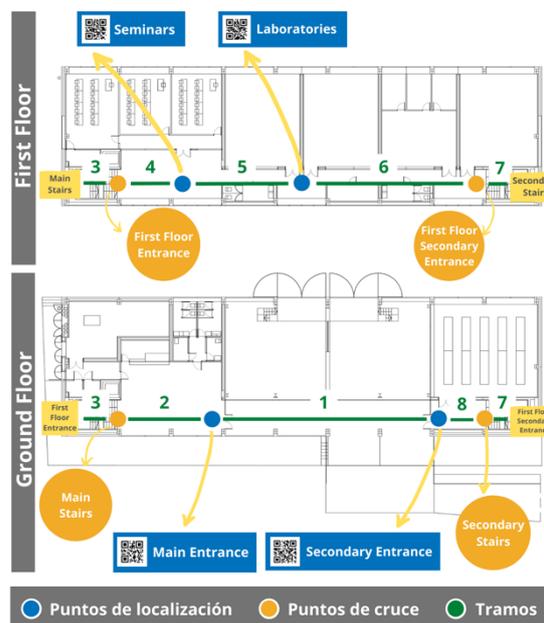


Figura 3. Ejemplo de mapa de un edificio con puntos de localización, puntos de cruce y tramo.

Así, para simplificar la programación, se ha decidido crear una división de la escuela en *tramos*. Éstos se definen por *puntos de cruce*, que son puntos relevantes del colegio en los que coincide más de un recorrido, o por *puntos de localización*. En consecuencia, las rutas posibles son el resultado de la unión de tramos. Para aclarar esta representación, la Figura 3 muestra el plano de un edificio de ejemplo en la UDC. Se trata de un edificio de dos plantas, en el que se han definido cuatro puntos de localización (marcados en color azul en la Figura 3): entrada principal (Main Entrance), entrada secundaria (secondary entrance), seminarios (seminars) y laboratorios (laboratories). Además, se han establecido cuatro puntos de cruce (marcados en color naranja en la Figura 3): escaleras principales (main stairs), escaleras secundarias (secondary stairs), entrada de la primera planta (first floor entrance) y entrada secundaria de la primera planta (first floor secondary entrance). En consecuencia, se han establecido los 8 tramos (16 si consideramos el sentido) que se muestran en la Figura 3 en color verde.

El conjunto de puntos de localización, puntos de cruce y tramos de recorrido conforman la representación topológica del mapa que se utilizará aquí. Para simplificarlo, se propone utilizar un *grafo*, que reduce el mapa 2D (plano) a un gráfico basado en nodos y enlaces como el que se muestra en la Figura 4. A partir de esta representación en grafo, es fácil guiar al usuario en la aplicación, como veremos más adelante.

Por lo tanto, la *principal tarea que tienen que llevar a cabo los alumnos es desarrollar este grafo*, similar

al de la Figura 4, para su colegio particular. Para ello, se recomienda partir de un plano del colegio similar al mostrado en la Figura 3, si es posible, y realizar el proceso "a mano" (en un papel impreso o utilizando una tableta), para que el concepto de cambio de representación quede más claro para los alumnos. Como se puede observar en la Figura 4, es necesario explicar a los alumnos que deben etiquetar los puntos con números. De esta manera, la representación se hace más sencilla para ser utilizada posteriormente en los pasos de razonamiento, ya que la programación es más eficiente si podemos utilizar números.

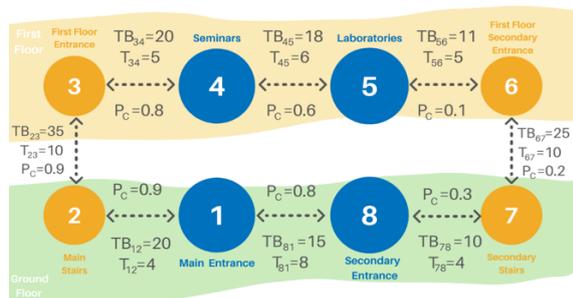


Figura 4. Representación como grafo del mapa topológico del edificio de ejemplo (los valores que aparecen en los enlaces se explicarán más adelante).

Para finalizar esta primera tarea, *los alumnos tienen que tomar fotografías del inicio y el final de cada tramo* (puntos de localización y puntos de cruce) con su *Smartphone*, que se mostrarán en su versión de la *app School Path Guide*, de forma similar a las mostradas en el vídeo "SchoolPathGuide.mov".

En la segunda tarea de la actividad 1, se propone que los alumnos comiencen a programar en App Inventor. En concreto, desarrollarán la representación de cada una de las secciones de la *app*, para que la *Pantalla1* y la *Pantalla2* funcionen correctamente, aunque todavía no se hayan calculado la ruta óptima. En concreto, los pasos que deben seguir son:

1. Actualizar el nombre de los puntos de localización en la plantilla del proyecto.
2. Subir las imágenes al proyecto de App Inventor.
3. Programar una función que muestre, en la *Pantalla2*, la información que se debe mostrar para orientarse en cada sección de la ruta. En concreto, se propone que finalicen la función llamada `Print_Sections()` (mostrada en la Figura 5) que está en la plantilla. Esta toma como entrada dos parámetros `section_origin` y `section_destination`, y llama a otra función que realiza la impresión en pantalla. Los alumnos tienen que programar una sencilla estructura `if-else` que compruebe el número de los puntos de origen y destino, según lo establecido en el grafo. Cada uno de estos condicionales llama a la función que imprime, `Print_origin_and_destination_images_names()`,

que recibe 5 parámetros de entrada (`imagen_origen`, `nombre_origen`, `imagen_destino`, `nombre_destino` e `indicaciones`), correspondientes a la información incluida en los nodos y enlaces de la representación del grafo. Esta función está pre-programada y simplemente deben usarla para que muestre los parámetros en las posiciones correctas de la *Pantalla2*.

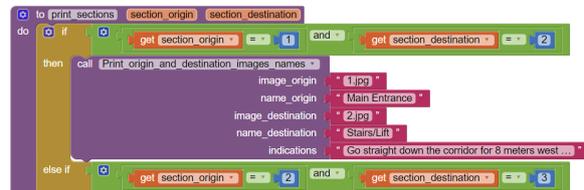


Figura 5. Extracto de la función `Print_Sections()` que deben finalizar los estudiantes.

3.2. Actividad 2: razonamiento

Esta actividad se centrará en introducir el concepto de razonamiento en IA desde un enfoque práctico. En concreto, los estudiantes conocerán los fundamentos del razonamiento probabilístico [12], que se trabajará sobre el grafo creado en la actividad anterior (Figura 4). Se puede observar en el mismo que, en este caso simple, sólo hay dos caminos posibles para cada ruta. Aunque se podrían utilizar otros criterios, se propone aquí considerar que la mejor ruta será la más rápida (mínimo tiempo requerido para recorrerla).

El primer paso que deben realizar los alumnos en esta tarea es calcular el tiempo necesario para recorrer cada tramo en su centro educativo (T_{ij} , donde i es el origen y j el destino). Se les recomienda obtener estos datos de forma empírica, midiendo los tiempos necesarios para recorrer el tramo caminando, aunque también se puede utilizar un valor aproximado. Estos valores de tiempo deben incluirse en los enlaces del grafo, como se muestra en la Figura 4, y corresponden al tiempo real en segundos necesario para atravesar cada tramo. Con esta representación, para calcular la ruta óptima es necesario sumar los tiempos de cada tramo para las dos rutas posibles, y compararlos.

La programación a realizar por los alumnos en esta tarea *implica crear una función que calcule la ruta óptima desde un origen a un destino* (etiquetado por su número) sumando los tiempos de cada uno de los tramos que constituyen la ruta. El resultado de esta función debe ser el sentido de la ruta (en el sentido de las agujas del reloj o en sentido contrario). Un algoritmo sencillo que realiza este cálculo utilizaría dos bucles, uno que se ejecuta en el sentido de las agujas del reloj (aumentando el número de nodo) y el otro en sentido contrario (disminuyendo el número de nodo). Una vez obtenida la dirección, se puede utilizar la función pre-programada disponible en la plantilla, llamada `route_representation()`, que recibe dicha dirección y muestra la información adecuada de los

tramos de la ruta, evitando que los estudiantes empleen mucho tiempo en el interfaz y su aspecto.

Con la realización de la tarea anterior, la aplicación ya resuelve el problema básico de selección de la ruta óptima, y puede ser probada por los alumnos. Pero no se puede considerar "inteligente", ya que se realiza un razonamiento muy simple, básicamente un proceso de búsqueda que no cambia. En la tarea final, se introduce información probabilística en el grafo, para que el razonamiento sea más interesante.

Para ello, consideramos una situación realista derivada del horario de recreo del colegio. En este periodo, algunos caminos pueden estar saturados, afectando al tiempo de recorrido mostrado en la Figura 4, principalmente aquellos que implican el uso de escaleras. Para gestionar esta situación, hay que incluir una nueva variable, la hora actual, que se puede obtener del reloj del *Smartphone* mediante App Inventor. Además, los estudiantes tienen que incluir dos nuevas variables en cada enlace del grafo de la Figura 4. La primera es la *probabilidad de congestión* (P_c), un valor entre 0 y 1 que representa la probabilidad de encontrar personas en un determinado tramo. Esta probabilidad es 0 fuera del recreo, pero hay que calcularla para cada tramo durante el mismo. Para ello, los estudiantes pueden utilizar un valor aproximado, o pueden estimarlo de nuevo empíricamente. Esta segunda opción es muy interesante para comprender el concepto de probabilidad de un suceso. Por ejemplo, podrían observar si los diferentes tramos están congestionados o no durante el recreo en diferentes días, y calcular un valor promedio. Si un cierto tramo estuvo congestionado 3 de 5 días, su probabilidad será de 0.6.

La segunda variable a incluir en el grafo es el *tiempo medio necesario para recorrer cada sección durante el recreo* (T_{Bij}). Este valor será mayor que el original T_{ij} , aunque puede tener un incremento diferente según el tramo. Para calcularlo empíricamente, los estudiantes deben recorrer las diferentes secciones en muchos periodos de congestión y calcular un tiempo medio. Los valores de estas variables P_c y T_{Bij} se muestran en la Figura 4 a modo de ejemplo para el caso del edificio de la UDC. Podemos observar que, al pasar de la planta baja a la primera planta (puntos 2 a 3) utilizando las escaleras, la probabilidad de encontrar congestión es de 0,9, lo que significa que el 90% de las veces no se alcanzará el tiempo normal de esta

sección T_{23} (10s), y se deberá utilizar T_{B23} (30s).

Una vez definidos estos nuevos valores por los alumnos, tienen que incluirlos en el programa mediante dos listas. El último paso que tienen que llevar a cabo es crear una función que lea la hora del reloj y la compare con la hora de descanso predefinida de su escuela. Si el usuario no está dentro de la hora de recreo, el cálculo de la ruta desarrollado en la tarea anterior no se modifica. Si está dentro del rango, entonces el tiempo que se utilizará en cada sección de la ruta será T_{ij} o T_{Bij} dependiendo de la probabilidad de congestión. Para calcularlo, los alumnos pueden crear un número aleatorio entre 0 y 1, y comprobar si está por encima o por debajo de la probabilidad de congestión. Estos cambios aplican a la función antes creada (*route_calculate()* en la plantilla), de manera similar a la mostrada en la Figura 6.

Finalmente, los estudiantes deben probar el funcionamiento adecuado de su aplicación con todas las características añadidas. Para ello, se les recomienda usarla ellos mismos, y también compartirla con otros alumnos que hagan de usuarios reales.

4. Resultados

Como se comentó en la sección de introducción, cada UD del proyecto AI+ es probada por los alumnos de los 6 centros participantes. De media, 6 profesores y 36 alumnos validan cada UD antes de ser puesta a disposición de la comunidad educativa. En [13] se incluye una carpeta denominada "feedback&results" donde se han añadido dos tipos de resultados asociados a la actividad aquí presentada. Por un lado, se han incluido los informes de realimentación de los profesores, y por otro, algunas soluciones creadas por los alumnos a modo de ejemplo.

De la realimentación dada por los docentes se han derivado varias mejoras en la UD inicial. Por ejemplo, el tiempo asignado para su realización en el aula fue incrementado de 6 a 8 horas. Otro de los problemas detectados y que está reflejado en los informes [13] fue que la parte de programación inicial era demasiado compleja, por lo que se decidió incluir en la plantilla más bloques preprogramados. Concretamente, aquellos que no tocan ningún tema de IA y son específicos de la creación de apps móviles, como la gestión de los eventos, o el aspecto estético de las pantallas.

```

set random_number to random fraction
if
  get global rush_hour = 1 and
  get random_number <= select list item list get global section_probability index get counter_aux
then
  set time_route_1 to get time_route_1 + select list item list get global section_time_rush_hour index get counter_aux
else
  set time_route_1 to get time_route_1 + select list item list get global section_time index get counter_aux

```

Figura 6. Extracto de la función *route_calculate()* que deben finalizar los estudiantes.

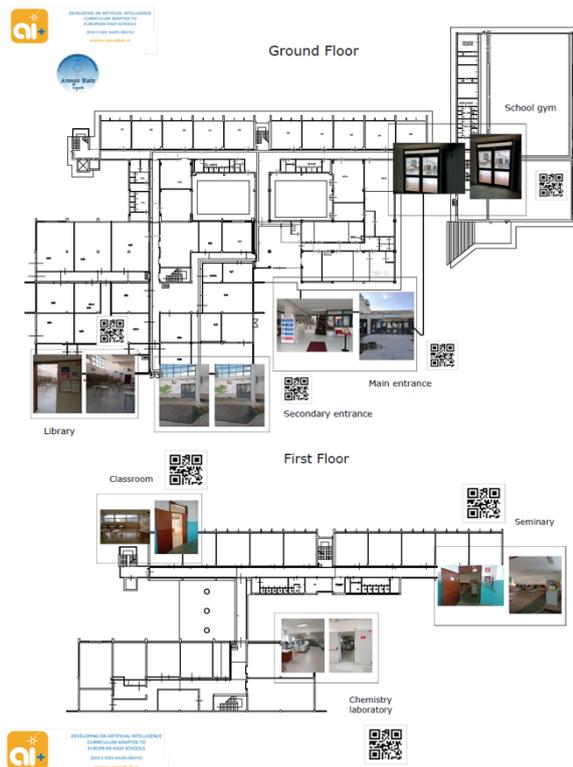


Figura 7. Mapa creado por un grupo de alumnos.

En cuanto al aprendizaje del alumnado, todos los profesores consideraron que el nivel de conocimientos adquiridos fue adecuado. No solo fueron capaces de realizar la aplicación y obtener un producto “real”, sino que lograron comprender los conceptos de representación y razonamiento en el campo de la IA. En la Figura 7 se muestra un ejemplo de mapa creado por un grupo de alumnos del centro italiano, mientras que otras soluciones creadas por los alumnos se pueden consultar en la carpeta de recursos de la unidad [13].

5. Conclusiones

Este artículo describe un recurso educativo para profesores de informática de la etapa pre-universitaria, interesados en introducir a sus estudiantes en el campo de la Inteligencia Artificial con un enfoque aplicado. Está disponible para su uso libre en la web del proyecto, e incluye un documento guía para el profesor con una descripción detallada de las actividades y tareas. Además, incluye las soluciones a los retos de programación propuestos, y una plantilla para que los estudiantes puedan tomar como base.

La principal conclusión obtenida tras su implementación ha sido que estudiantes sin conocimientos previos sobre representación del conocimiento y razonamiento en el marco de la IA, fueron capaces de comprender claramente estos conceptos y, además, de desarrollar una aplicación para *Smartphone* realista.

Referencias

- [1] AIplus project, Accedido el 11/02/2022. <http://aiplus.udc.es>
- [2] AI4K12 initiative, Accedido el 11/02/2022. <https://ai4k12.org>
- [3] CBSE Artificial Intelligence Curriculum Launch. Accedido el 11/02/2022. <http://cbseacademic.nic.in/ai.html>
- [4] European Commission. Coordinated Plan on Artificial Intelligence (COM (2018) 795 Final). Accedido el 11/02/2022. <https://bit.ly/3hdLbsg>.
- [5] European Commission. Digital Education Plan 2021-2017, accedido el 11/02/2022. <https://education.ec.europa.eu/es/plan-de-accion-de-educacion-digital-2021-2027>
- [6] M. Gelfond, Y. Kahl. Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach. Cambridge University Press, 2014.
- [7] S. Guerreiro-Santalla, F. Bellas, R.J. Duro. Artificial Intelligence in Pre-University Education: What and how to teach. Proceedings 3rd XoveTIC, vol. 54, pp. 48. MDPI, 2020.
- [8] Home Assistant. Accedido el 11/02/2022. <https://www.home-assistant.io>
- [9] Jeremy Knox. Artificial intelligence and education in China, *Learning, Media and Technology*, 45:3, 298-311, 2018
- [10] M. Khairiyah, A. Syed, Z. Mohammad, F. Nor. Cooperative Problem-based Learning (CPBL): Framework for Integrating Cooperative Learning and Problem-based Learning. *Procedia - Social and Behavioral Sciences* 56: 223-232, Elsevier. 2012.
- [11] MIT App Inventor. Accedido el 11/02/2022. <https://appinventor.mit.edu/explore/ai-with-mit-app-inventor>.
- [12] J. Pearl. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann. 2014.
- [13] Recurso educativo UD3 (AI+ project) – Representation and Reasoning. Accedido el 11/02/2022. <https://cutt.ly/ckKKE1N>
- [14] S. Russell, P. Norvig. Artificial Intelligence. A Modern Approach. Pearson. 2021.
- [15] The Robobo Project. Accedido el 11/02/2022. <https://theroboboproject.com>
- [16] D. Touretzky, C. Gardner-McCune. Artificial Intelligence Thinking in K-12, to appear in *Computational Thinking Education in K-12: Artificial Intelligence Literacy and Physical Computing*, chapter 8. The MIT Press, 2022.
- [17] Xunta de Galicia. DOGA con la publicación oficial de la nueva materia de IA. Accedido el 11/02/2022. <https://cutt.ly/fO4Rvlg>