UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

TALLER REMOTO BASADO EN APPINVENTOR PARA PROMOVER EL DESARROLLO DEL PENSAMIENTO COMPUTACIONAL EN NIÑAS Y NIÑOS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERA CIVIL EN COMPUTACIÓN

KARINA ANDREA PARGA VALENZUELA

PROFESOR GUÍA:
FRANCISCO GUTIÉRREZ FIGUEROA

MIEMBROS DE LA COMISIÓN:
MARÍA CECILIA BASTARRICA PIÑEYRO
PATRICIO INOSTROZA FAJARDIN

Este trabajo ha sido parcialmente financiado por Proyecto FONDECYT Nº 11190248

SANTIAGO DE CHILE
2021

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERA CIVIL EN COMPUTACIÓN
POR: KARINA ANDREA PARGA VALENZUELA
FECHA: 2021
PROF. GUÍA: FRANCISCO GUTIÉRREZ FIGUEROA

TALLER REMOTO BASADO EN APPINVENTOR PARA PROMOVER EL DESARROLLO DEL PENSAMIENTO COMPUTACIONAL EN NIÑAS Y NIÑOS

El pensamiento computacional es una habilidad que se ha vuelto útil en una gran cantidad de disciplinas y en la vida diaria. Por esta misma razón, en estos últimos años se ha querido masificar esta habilidad en la población, especialmente en el nivel escolar. Actualmente, el Departamento de Ciencias de la Computación ofrece el taller "Desarrollando el Pensamiento Computacional: Nivel II", donde se busca mostrar se busca mostrar la computación a niñas y niños de quinto a octavo básico como una rama de la ciencia en la cual se pueden crear aplicaciones útiles e interactivas a través de la programación. Al mismo tiempo se busca inculcar la idea a las y los estudiantes que la programación y la computación es algo que todas y todos pueden hacer.

En este trabajo de título se rediseña el taller existente en dos ámbitos: primero rediseñando el curso en torno a las trayectorias de aprendizaje propuestas por Rich et al. y luego adaptando el material a una modalidad de clases remota. Esto, para adaptarse a la situación actual de enseñanza a distancia y para hacer uso de propuestas recientes en el área de la enseñanza del pensamiento computacional.

Para ello, se realizó un estudio de la literatura y un análisis del taller original. Luego, se diseñó una planificación para el taller siguiendo los lineamientos brindados por la literatura. Posteriormente se ajustó el taller a un formato de cuatro días, adecuándolo a la planificación escolar actual de las niñas y niños. El taller se desarrolló bajo la tutela del Departamento de Ciencias de la Computación entre el 29 de mayo y el 19 de junio de año 2021, todos los sábados a las 10 de la mañana.

Para validar el taller, se llevó a cabo un análisis siguiendo la metodología de caso de estudio propuesta por Yin. Esto significa plantear preguntas que el taller debe responder y recolectar la evidencia necesaria entregada por el modelo de caso de estudio. Se analizaron las respuestas que da la experiencia del taller a la luz de distintos trabajos relacionados a la enseñanza de pensamiento computacional, en el rango de quinto a séptimo básico, dado las características de los asistentes.

Este trabajo presenta un primer acercamiento a la enseñanza del pensamiento computacional siguiendo el modelo de trayectorias de aprendizaje en Chile, y en formato de curso extra-programático en modalidad remota. En este trabajo se concluyó que el uso de las trayectorias de aprendizajes sugeridas logró que los asistentes incorporasen conceptos de pensamiento computacional en un contexto de enseñanza remoto.

Para todos a quienes se les acabó el camino que habían tomado: hagan uno nuevo, de cero, propio. No te rindas. Y feliz cumpleaños Benja

Agradecimientos

Gracias a mi mamá, mi papá y mi hermana, por soportarme estos años de tropiezos y lentitud. Les quiero mucho más de lo que alcanzo a decir.

Gracias a Benja y Franco por acogerme en los últimos meses de mi memoria y darme el refugio que necesitaba.

Gracias a Nancy, a Jocelyn y a Cecilia Rivara por creer en mi antes que yo misma.

Gracias a Elisa por mostrarme que se puede ser genial sin ser tryhard.

Gracias a Constanza por inspirarme.

Gracias a Memo por cuidarme por tanto tiempo.

Gracias a la salita y la ofisalita: mi paso por la u no hubiese sido ni la mitad de memorable y acogedor sin todos con los que me encontré ahí.

Gracias a Patricio por dejarme tomar notas de su taller y así partir el mio con el pie derecho.

Gracias especiales a Tomimi, Mati, Franquito de nuevo, Cristóbal y Witu por ayudarme a revisar el Kahoot.

Gracias a Francisco, por guiarme y apoyarme durante toda la memoria.

Gracias a Sebastián, Emilio y Bryan: el taller no hubiese sido posible sin ustedes, en el sentido más práctico de la palabra. Y gracias por creer en mi taller.

Gracias a Bichito, por darme un espacio para conversar y cooperar en torno a la educación en niños.

Y gracias a todos los que conocí en mi paso por la facultad: les tengo a todos un pedacito de mi corazón con su nombre.

Este trabajo de tesis ha sido parcialmente financiado por el proyecto Fondecyt Nº 11190248.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

El pensamiento computacional es la habilidad de abstracción y descomposición cuando se aborda una tarea compleja de gran tamaño o se diseña un sistema complejo de gran tamaño, para que luego las partes resultantes puedan ser manejadas por un computador [22]. Por ejemplo, si evaluamos el problema de cómo separar frutas por peso de manera algorítmica, , la solución obtenida podrá ser ejecutada por un objeto programable. En el año 2006, Jeannette M. Wing [22] plantea la importancia de esta manera de resolver problemas y propone que sea incorporada a los planes curriculares de las escuelas. Esto es debido a que el pensamiento computacional es una herramienta apta para problemas científicos y especialmente relevante en la medida que adoptamos más tecnologías en nuestra cotidianeidad [14].

El taller "Desarrollando el Pensamiento Computacional: Nivel II" se dicta a niñas y niños de entre quinto y octavo básico (siendo este nivel educacional un subconjunto del denominado como K-8 en Estados Unidos, que abarca desde kinder hasta octavo básico) en las dependencias del Departamento de Ciencias de la Computación (DCC) de la Facultad de Ciencias Físicas y Matemáticas (FCFM) de la Universidad de Chile. El taller dura una semana, con clases diarias de 3 horas y una presentación final. Su objetivo es mostrar la computación como una rama de la ciencia donde se pueden crear aplicaciones útiles e interactivas a través de la programación, al mismo tiempo que mostrar a las y los estudiantes que la programación y la computación es algo que todas y todos pueden hacer. En el taller se presentan conceptos de Programación e Ingeniería de Software como: arquitectura Modelo-Vista-Controlador (MVC), desarrollo guiado por patrones, programación de a pares, testeo y *debugging*, así como conceptos de pensamiento computacional, tales como variables, secuencias, ciclos, condiciones, entre otros, de manera didáctica y cohesionada. La primera versión del curso fue realizada en el año 2018 a modo de piloto para futuras versiones.

En el año 2020, motivado por la situación mundial de pandemia, las clases a nivel nacional se han adherido a un modelo no presencial a fin de evitar riesgos de contagio por exposición y contacto directo. Esto también se ha replicado en distintas iniciativas educativas, como charlas y conferencias, lo cual ha brindado la oportunidad de asistir a estas instancias sin reuniones innecesarias de personas.

Dado que la primera edición del taller "Desarrollando el Pensamiento computacional: Ni-

vel II" fue una versión piloto, su estructura y elementos aún están en un proceso de prueba. Considerando esto y las últimas propuestas sobre enseñanza de pensamiento computacional (basadas en trayectorias de aprendizaje, que es un mapa de progreso en el tiempo del desarrollo lógico y armónico de una competencia), se plantea como relevante la implementación de estas ideas para actualizar el taller en torno a ellas y proveer experiencias que aporten a al desarrollo y avance mediante evidencia de dicha propuesta. Además, dado que el Departamento de Ciencias de la Computación desea darle continuidad en este escenario a su labor de vinculación con el medio mediante el taller de Pensamiento Computacional, se aprovecha esta oportunidad para evaluar un formato remoto para los talleres que brinda, adecuado al contexto particular de Chile.

En vista de lo anterior, en el presente trabajo de título se evaluó la factibilidad de incentivar el aprendizaje del pensamiento computacional y desarrollo de aplicaciones de software en niñas y niños chilenos usando un currículo basado en trayectorias de aprendizaje para pensamiento computacional. El taller experimentó con una modalidad remota para disponibilizarlo de mejor manera, probando distintos aspectos de las clases *online* a fin de hacerlo más llevadero para los alumnos y el cuerpo docente.

1.1. Problema abordado

Para contextualizar, una trayectoria de aprendizaje es definida por Clements y Sarama [6] como un modelo pedagógico que describe una secuencia de tareas, actividades y objetivos, para lograr el desarrollo de una o más competencias específicas. De momento existen trayectorias de aprendizaje propuestas desde el MINEDUC para las asignaturas de Lenguaje y Comunicación, Historia, Geografía y Ciencias Sociales, Matemática y Ciencias Naturales (de 1° a 6° básico)[1]. Sin embargo, desarrollar trayectorias de aprendizaje se basa en la observación de estrategias de enseñanza y sus resultados, por lo que es un trabajo iterativo de ensayo y error. En el año 2017, K. Rich, C. Strickland, T. Binkowski, C. Molan y D. Franklin [17] proponen tres trayectorias de aprendizajes orientadas al desarrollo del pensamiento computacional: secuencia, repetición y condicionales, las cuales se derivan del análisis de la literatura disponible hasta ese momento.

Por otro lado, App Inventor[2] es un entorno de programación visual para programación en bloques desarrollado por Google Labs, pensado para crear aplicaciones que corran en el sistema operativo Android. Las diferencias que presentan los lenguajes de programación en bloques, comparado con los lenguajes de programación escritos, son la encapsulación de varias instrucciones en un solo objeto y la guía visual de la posición de ciertos elementos con respecto a otros mediante encajes como si fueran piezas de un puzzle. Esto lo hace ideal para introducir la programación a gente que no ha tenido acercamiento a código de programas antes, que en este caso son niñas y niños de enseñanza básica.

El taller "Desarrollando el Pensamiento Computacional: Nivel II" hace uso de App Inventor

[1] `https://especial.mineduc.cl/recursos-apoyo-al-aprendizaje/recursos-las-los-docentes/progresiones-de-aprendizaje-en-espiral-y-orientaciones-para-su-implementacion/`

[2] `appinventor.mit.edu/`

y está originalmente diseñado con actividades como vehículo para transmitir a los asistentes los conceptos de programación a enseñar. Durante el curso se presentan desafíos y se guía a los alumnos en el desarrollo de la solución de paso enseñando conceptos relacionados a la herramienta App Inventor. Los desafíos suben en complejidad a medida que avanzan las clases, para culminar en el desarrollo de una aplicación de su propia autoría. A nivel de enseñanza de la computación (como es un área reciente), es importante mantenerse actualizado con las ultimas investigaciones y es por eso que surge la iniciativa de rediseñar el taller en torno a las propuestas educacionales más recientes , que plantean a las trayectorias de aprendizaje como punto de partida para el pensamiento computacional .

Adicionalmente, el taller está pensado para implementarse presencialmente, porque el equipo a cargo posee las *tablets* y computadores necesarios para desarrollar las actividades y se vale de las instalaciones del DCC de la FCFM para tener un espacio seguro donde interactuar con menores de edad. Sin embargo, dada la situación actual de confinamiento, resulta interesante desde un punto de vista práctico y como objeto de investigación en el área de Educación en Ciencias de la Computación, el explorar cómo una modalidad remota del curso podría funcionar. Por lo tanto, el problema es que actualmente el taller tiene un diseño que no se ajusta a las trayectorias de aprendizaje propuestas para enseñar pensamiento computacional y tampoco tiene la estructura necesaria para ser llevado a cabo de manera remota.

La literatura existente sobre casos similares al taller de App Inventor se divide en varios grupos:

- Instancias educativas en computación a nivel escolar en contextos de talleres informales con un currículo orientado a actividades, tales como campamentos intensivos de verano [19], horas del código [2], talleres de programación competitiva[3] [4] o clases particulares de un lenguaje de programación en particular [5][6]. Éstas buscan habilitar a los estudiantes en la programación, pero no necesariamente siguiendo un modelo de trayectorias de aprendizaje para abordar el pensamiento computacional. Esto es importante porque, según Clements y Sarama [6], el conocimiento se adquiere mejor cuando se usa un enfoque constructivista para estructurar el currículum, haciendo uso de lo previamente aprendido para profundizar o relacionar los nuevos conceptos. Esto nos lleva a concluir que los conceptos entregados en los talleres podrían presentarse de mejor manera.
- Instancias educativas de pensamiento computacional integradas al currículo escolar, tanto en el formato de clases a niñas y niños, como con talleres a docentes para habilitarlos en el desarrollo de actividades que apunten a los objetivos planteados por las trayectorias de aprendizajes. Este tipo de material provee un acercamiento a qué actividades resuenan mejor con los estudiantes en el contexto de su edad y habilidad de programación y resolución de problemas en general, pero la duración de la intervención difiere de lo que puede lograr el taller actualmente.
- Instancias educativas que hacen uso de trayectorias de aprendizaje y que están orientados a los primeros años de universidad en carreras de ciencia o de programación. Si

[3] `https://ninaspro.cl/#/incentiva`
[4] `http://www.olimpiada-informatica.cl/acercade`
[5] `https://www.jovenesprogramadores.cl/que-aprenderas/`
[6] `https://desafiolatam.com/taller-programacion-para-ninos/`

bien tanto en el caso del taller como en los que presentan la literatura los participantes son programadores novatos, los paralelos que se pueden hacer se ven limitados por la diferencia de edad (9 a 11 años en contraste con 17 o más), que implica un mayor conocimiento de estrategias de resolución de problemas, y que el contexto universitario es distinto al escolar.

Ninguno de los trabajos anteriormente mencionados integra los elementos importantes sobre el caso que se propone estudiar en este trabajo de memoria, por lo que no es posible tomar una solución ya hecha e implementarla: se debe desarrollar un programa en base a las trayectorias de aprendizaje en pensamiento computacional (surgidas el año 2017 por Rich et al. en la Universidad de Chicago [17]) para poder evaluar su efectividad en la práctica, añadiéndole también la particularidad de un contexto remoto. Este trabajo de memoria es la puesta a prueba de un taller diseñado en torno a trayectorias de aprendizaje para el área de pensamiento computacional, lo que aporta valor a la investigación en esa área . Adicionalmente, el contexto donde se efectuaron las pruebas fue remoto, por lo que los aprendizajes obtenidos benefician a los demás talleres de pensamiento computacional que se desarrollen en condiciones similares; esto es, en formato remoto y en Chile.

1.2. Objetivos

A continuación se presentan los objetivos para la ejecución del trabajo de título propuesto.

1.2.1. Objetivo general

El objetivo general de esta memoria es *estudiar la efectividad de enseñar pensamiento computacional y nociones de programación a niñas y niños chilenos en el rango de K-8, bajo un diseño de trayectorias de aprendizaje, de manera remota, valiéndose de App Inventor como lenguaje de programación de apoyo.*

1.2.2. Objetivos específicos

Para cumplir con lo anterior, se plantearon los siguientes objetivos específicos:

1. Identificar los parámetros relevantes para evaluar la efectividad de aprendizaje de pensamiento computacional en un contexto virtual.
2. Analizar y adaptar las plantillas de desarrollo, actividades de orientación, *snippets*, pautas de evaluación para el equipo docente y material de apoyo con las que cuenta el taller para adaptarlo al taller remoto .
3. Diseñar e implementar la estructura para dar soporte al taller, haciendo uso de herramientas probadas y validadas.

4. Estudiar la pertinencia de las actividades diseñadas en el marco del taller en la dimensión de actividad remota.
5. Estudiar la pertinencia de las actividades diseñadas en el marco del taller en términos de desarrollo de habilidades de pensamiento computacional .
6. Desarrollar una propuesta para la replicabilidad de este taller.

1.3. Solución desarrollada

Para cumplir con los objetivos anteriormente presentados, la solución desarrollada es un ajuste y rediseño del taller original, siguiendo recomendaciones actualizadas de la literatura (trayectorias de aprendizaje propuestas [17]), producción de insumos tales como guías y presentaciones para el cuerpo docente, *snippets* de código usados en las actividades, y un estudio de la aplicabilidad del taller a nivel de prototipo en un escenario de ejecución de manera remota.

La solución resuelve el problema planteado al crear un programa que ponga a prueba las propuestas de trayectorias de aprendizaje y evaluar los resultados, entregando información útil para la investigación de la enseñanza de pensamiento computacional en escolares, que es un área joven y en constante evolución.

Cada objetivo especifico es resuelto de la siguiente manera:

1. *Identificar los parámetros relevantes para evaluar la efectividad de aprendizaje de pensamiento computacional en un contexto virtual*: se resuelve mediante la investigación y análisis de trabajo relacionado.
2. *Analizar y adaptar las plantillas de desarrollo, actividades de orientación, snippets, pautas de evaluación para el equipo docente y material de apoyo con las que cuenta el taller para adaptarlo al taller remoto*: se resuelve en la fase de análisis y diseño, reformulando el material existente en torno a las sugerencias de la literatura actual.
3. *Diseñar e implementar la estructura para dar soporte al taller, haciendo uso de herramientas probadas y validadas*: se resuelve llevando a cabo el taller rediseñado.
4. *Estudiar la pertinencia de las actividades diseñadas en el marco del taller en la dimensión de actividad remota*: se resuelve analizando la información recolectada y contrastando con la literatura relacionada.
5. *Estudiar la pertinencia de las actividades diseñadas en el marco del taller en términos de desarrollo de habilidades de pensamiento computacional*: se resuelve con instrumentos como cuestionarios, análisis estático de código y evaluación cualitativa por rúbricas [11].
6. *Desarrollar propuestas para la replicabilidad de estos talleres*: se resuelve compilando las acciones que lograron su objetivo en el taller, los motivos por los cuales funcionaron y cómo se lleva a la practica en distintos casos.

1.4. Metodología

La solución fue desarrollada siguiendo un enfoque basado en incrementos de diseño-implementación del material del taller, con una evaluación a nivel piloto con una muestra controlada de participantes, miembros de la población objetivo. El análisis de la solución se guía por el marco de un caso de estudio, propuesto por Robert K. Yin [23].

1.5. Estructura del documento

En este documento, se presenta y discute sobre el trabajo relacionado en el capítulo 2. Luego, en el capítulo 3, se procede a ahondar el análisis previo y en las decisiones de diseño de la propuesta. En el capítulo 4 se discuten los aspectos operacionales de la implementación de la solución, poniendo en evidencia las consideraciones técnicas tomadas para el desarrollo del taller. En el capítulo 5 se validó la implementación haciendo uso de la metodología de caso de estudio. En el capítulo 6 se discuten los resultados a la luz de los trabajos relacionados, se elaboran recomendaciones para replicar la experiencia. Finalmente, en las conclusiones se compila el conocimiento adquirido y el aporte generado, y se propone trabajo futuro.

Capítulo 2

Trabajo Relacionado

Varios autores y publicaciones han abordado la idea de la enseñanza del pensamiento computacional en niñas y niños, pues es un campo de estudio interesante y actual. Previo a este trabajo, se realizó una revisión informal de artículos publicados en las conferencias SIGCSE (*ACM Technical Symposium on Computer Science Education*), ITiCSE (*ACM International Conference on Innovation and Technology in Computer Science Education*), ICER (*ACM International Conference on Computing Education Research*) y WiPSCE (*International Workshop in Primary and Secondary Computer Science Education*) entre los años 2015 y 2020. Las conferencias mencionadas son las líderes en el área de Educación en Ciencias de la Computación y forman el cuerpo de estudio más relevante y actualizado que puede ser usado en este trabajo de título. Inicialmente se seleccionaron los *papers* por título, clasificándolos posteriormente de la siguiente manera:

- *Relevante*: Las *keywords* y *abstracts* mencionan directamente intersecciones entre App Inventor, trayectorias de aprendizaje, pensamiento computacional y enseñanza en K-8. La importancia de estas *keywords* está en que son los tópicos centrales de esta memoria.
- *Posiblemente útil*: Las *keywords* y *abstracts* hablan de entornos de desarrollo visual en general, elementos de pensamiento computacional, enseñanza en K-12, instrumentos de evaluación, aspectos de la pedagogía y psicología de enseñanza de pensamiento computacional y programación a nivel escolar.
- *Complementario*: Las *keywords* y *abstracts* hablan de la metaenseñanza de pensamiento computacional, inserción en el medio, observaciones empíricas en situaciones especificas, entre otros.
- *No relevante*: Las *keywords* y *abstracts* tienen relación con la enseñanza de pensamiento computacional, pero está demasiado alejado de las otras áreas de interés.

Luego, dentro de aquellos artículos catalogados como relevantes, se estudiaron y seleccionaron aquellos con mayor cercanía al tema a tratar en este trabajo de titulo. Esto se traducen en seleccionar aquellas que donde se encontrasen al menos 2 de las *keywords* mencionadas.

Dentro de las investigaciones más relevantes para este trabajo, se puede mencionar a Shuchi Grover y Satabdi Basu, autoras del *paper "What We Can Learn About Student Learning From Open-Ended Programming Projects in Middle School Computer Science"* [10], que en el año

2018 develaron patrones de uso por parte de estudiantes de cursos de ciencia computacional que usaban Scratch y App Inventor. En este artículo, las autoras parten de la premisa que los proyectos hechos en Scratch y App Inventor en un régimen de creación libre son una fuente de información muy importante sobre qué logran realmente aprender las niñas y niños acerca de programación y pensamiento computacional y proponen que vale la pena analizarlos. Para ello, desarrollaron rúbricas que evalúan múltiples aspectos (tales como la cantidad de errores de programación, la evaluación de correctitud al programar, el tipo de aplicativo que desarrollan) de cerca de 80 proyectos seleccionados al azar. Gracias a esto, se descubrieron patrones en los tipos de artefactos programados y las estrategias más y menos usadas, tanto a nivel global como clasificados por género, curso y experiencia de la persona que guiaba los talleres. El estudio sienta precedentes sobre cómo abordar estudios de esta naturaleza, que es el caso de la presente memoria.

Otro *paper* fundamental para este trabajo es *"K–8 Learning Trajectories Derived from Research Literature: Sequence, Repetition, Conditionals"* [17], donde los autores investigan la literatura actual sobre enseñanza de pensamiento computacional y proponen tres ejes centrales para su aprendizaje efectivo, con distintos niveles de acercamiento al tema por parte de los estudiantes. Primero, las trayectorias de aprendizaje son un mapa de progreso en el tiempo del desarrollo lógico y armónico de una competencia, usadas por educadores de diversas áreas para planificar de manera coherente un curso de larga duración. Eso hace que sea deseable integrarlas en todos los planes de enseñanza de pensamiento computacional. Sin embargo, el estudio sobre el desarrollo del pensamiento computacional y programación en edades tempranas de manera efectiva y eficiente es muy reciente, por lo que aún deben ser descubiertas esas trayectorias. Mediante la investigación exhaustiva de *papers* en el área, se logra llegar a la propuesta de tres trayectorias cruciales para el pensamiento computacional: secuencia, repetición y condiciones. El estudio propone un modelo espiral de enseñanza (un tópico se ve con creciente profundidad a medida que pasa el tiempo y se estudian variados tópicos en un periodo) fuertemente inspirado por las trayectorias de aprendizaje usadas en la enseñanza de matemáticas, sin embargo carece de implementación. El taller de App Inventor está diseñado en base a las trayectorias propuestas en este *paper*, por lo cual sería un aporte directo de evidencia sobre su efectividad.

También se analizó la publicación titulada *"Is Drawing Video Game Characters in an Hour of Code Activity a Waste of Time?"* [2], que estudia el nivel de compromiso, participación y absorción de conocimientos, guiándose por el tiempo dedicado a personalizar un personaje que es parte de la historia. Este *paper* se sitúa en una actividad de corta duración (una hora), donde se motiva a participantes sin experiencia previa en computación a programar parte de un código por primera vez. Esto se logra haciendo uso de código desarrollado previamente en lenguajes de programación de bloque. Se usa este tipo de lenguajes de programación porque tiene la particularidad de encapsular varias instrucciones en un solo bloque, el cual se posiciona con respecto a otros para programar. Lo anterior facilita el acercamiento a la programación, disminuyendo el tiempo de aprendizaje y desarrollo. El resultado del estudio fue que las personas que participan suelen tener un mayor compromiso con la actividad, y por ende, mejor retención del conocimiento, en caso de que puedan personalizar algo e interactuar con ello. Dado que la disciplina de enseñar tiene, en parte, un reto en la retención de entusiasmo, el contar con estrategias para manejarla en un contexto experimental como el que se propone es una herramienta imprescindible.

Con una visión distinta a los *papers* anteriores, en *"Designing for Integrated K-5 Computing and Literacy Through Story-making Activities"* [21] se propone que la programación y la narración pueden ir de la mano, algo muy distinto al vinculo casi inseparable entre matemática y computación que ha dominado las propuestas de enseñanza de pensamiento computacional. Se proponen actividades y rúbricas asociadas a ellas y se mide la retención de conocimiento al evaluar qué soluciones dieron a ciertos problemas. Estas rúbricas son una fuente de inspiración importante para la evaluación de las actividades del taller de App Inventor.

Otro autor relevante y más reciente es Janne Fagerlund, quien publicó el trabajo *"Computational thinking in programming with SCRATCH in primary schools: A systematic review"* [9], donde propone un currículo basado en Scratch para la educación básica en Finlandia. Se analizó formalmente la literatura para entender qué actividades cumplían el rol de fomentar el pensamiento computacional bajo el alero de un currículo enfocado a enseñar programación y se proponen métodos para hacerlo efectivo. Se identificaron actividades para Scratch que cultivan el desarrollo de pensamiento computacional para el nivel escolar K-9 y se desarrollaron rúbricas de evaluación que, si bien los autores plantean como inmaduras, son un buen punto de partida. El trabajo realizado y las conclusiones, como que la exploración de de métodos de enseñanza y diseño de actividades está en una etapa inicial, o que no se sabe la calidad del entendimiento que van desarrollando los alumnos mientras programan, son símiles a lo que se busca con el taller, pero a una mayor escala.

Con respecto a investigaciones que hablen sobre enseñanza en modalidad remota, se encontraron tres *papers* que se consideran útiles, pese a que se enfocan en estudiantes universitarios. En primer lugar, *"Factors for Success in Online CS1"* [5], donde se hace notar que el correcto desarrollo de actividades no evaluativas se relaciona con una adquisición efectiva de conocimiento. También mencionan que la autoeficacia (es decir, la confianza en la propia capacidad para lograr los resultados pretendidos) es clave para un aprendizaje acabado y que esta cualidad se ve reforzada por la presencia continua de profesores y mentores a lo largo de la clase remota. Esto guía el diseño del taller, donde se podría optar por una modalidad remota asíncrona, pero que sería sub-óptimo en vista de esta evidencia.

Luego está *"Choosing Face-to-face or Video-based Instruction in a Mobile App Development Course"* [4], donde se evidencia que, dada la oportunidad de elegir entre clases presenciales, híbridas o remotas, las y los estudiantes se sienten más cómodos con clases presenciales acompañadas de vídeos para poder repasar los contenidos que no lograron retener en las sesión: así logran complementar la inmediatez de las lecciones con la flexibilidad del auto-estudio. Lamentablemente, el estudio difícilmente puede ser generalizado dado que la muestra es muy pequeña y el caso es aplicado en un contexto donde se puede elegir libremente la modalidad. Este estudio nos muestra que es posible enseñar desarrollo de aplicaciones de manera remota y que los alumnos siempre van a valorar instancias de interacción en tiempo real con sus tutores, incluso si después hacen uso de material asíncrono.

Finalmente, se tiene *"Developing Soft and Technical Skills Through Multi-Semester, Remotely Mentored, Community-Service Projects"* [8], un estudio lleno de información muy útil para el manejo de proyectos de larga duración con programadores inexpertos. De los conocimientos más rescatables para el caso del taller está que el curso debe ser entendido como algo que no apasiona a todos los asistentes, y que por ende debe tener más estímulos que el

conocimiento de pensamiento computacional en sí mismo. Otro aprendizaje que proponen los autores es que resulta útil acotar los tiempos de trabajo, en tanto que da buenos resultados para un proyecto de larga duración y consigue que los mismos sean espacios productivos y de mucha interacción con los tutores. Este trabajo nos da dos lineamientos importantes: el primero es sobre generar más estímulos que la sola materia para los alumnos (para lograr un entusiasmo sostenido en el tiempo) y el segundo sobre la estructuración del tiempo, que debe ser explícito para cada actividad a desarrollar.

Como se puede notar, existen varios estudios sobre cómo aprenden las niñas y niños en ciertas condiciones, y también existen algunos estudios sobre enseñanza en una modalidad remota. No obstante lo anterior, en la literatura analizada no existe una observación en torno a instancias educativas dirigidas específicamente a enseñar pensamiento computacional, que sigan trayectorias de aprendizaje definidas y que se desarrollen remotamente. Es en esa intersección donde se sitúa el trabajo de título aquí presentado.

El taller de App Inventor, el objeto de trabajo de esta memoria, cuenta con un rediseño que hace uso de objetivos diarios y finales, basados en las trayectorias de aprendizaje descritas en [17], con actividades propuestas que apuntan a reforzar los objetivos diarios y finales. Las actividades fueron diseñadas teniendo en cuenta el aprendizaje obtenido en *"Is drawing video game characters in an hour of code activity a waste of time?"* y*"Designing for integrated k-5 computing and literacy through story-making activities"*, y están detalladas en el capitulo 4: Implementación.

En síntesis, el trabajo de título contribuye al área de Educación en Ciencias de la Computación, y más precisamente al desarrollo del pensamiento computacional en niñas y niños, mediante la evaluación *in situ* de las trayectorias de aprendizaje propuestas en *"K-8 learning trajectories derived from research literature: Sequence, repetition, conditionals"* en un contexto de aprendizaje remoto. Adicionalmente, se recolectan y consideran distintas técnicas y observaciones de los distintos trabajos revisados para aumentar la efectividad del taller y reducir el tiempo de diseño, pues es avanzar usando consideraciones ya probadas.

Capítulo 3

Análisis y Diseño de la Solución

En este capítulo se discute el análisis a la propuesta previa del taller y las decisiones de diseño que se tomaron para plantear una solución.

3.1. Análisis

En primera instancia, se hará un repaso de la estructura preexistente del taller y un análisis de la misma. El diseño original del taller de Pensamiento Computacional consiste en 5 clases, con 4 actividades de desarrollo de software que enseñan incrementalmente sobre el manejo de App Inventor y sobre nociones de programación y desarrollo de software. Adicionalmente, estos conocimientos son usados para el desarrollo de un proyecto propio de los asistentes, que será formulado en base a sus elementos y refinado siguiendo una metodología de Keep-Fix-Try[1].

El taller cuenta con un *handout* que describe las actividades que deben realizar los asistentes. Esta guía está separada en sesiones, con explicaciones para cada etapa de cada actividad. Además, cuenta con preguntas reflexivas sobre las actividades y un espacio asignado para que las respondan de manera escrita. Esta guía fue concebida como un apoyo a las clases, donde los alumnos pudiesen anotar sus observaciones y buscar información sobre las actividades y su desarrollo. Sumado a lo anterior, las actividades no vienen con código base, por lo que todo debe ser programado por los alumnos.

Tabla 3.1: Planificación del curso original por día

Día	Nombre de la actividad	Descripción	Objetivos
Clase 1	Introducción a App Inventor	Explicación de qué es App Inventor y cómo empezar a usarlo.	Familiarizarlos con el uso de la herramienta.

[1] `https://code-artisan.io/retrospective-method-kpt/`

	Actividad 1: Hello Purr!	Desarrollo guiado de aplicación sencilla donde diseñarán una interfaz de usuario (con botón, etiqueta y texto), programación del controlador (usando los bloques de sonido y botón) y empaquetado de la app.	Conceptos básico de programación en App Inventor. Concepto implícito de MVC
	Actividad 2: Adivina el número	Desarrollo guiado de aplicación donde la app genera un número aleatorio y el usuario debe adivinar qué número es. En la interfaz de usuario hay una etiqueta que dice "Número", un visor de campo de texto y un botón. Al final de la actividad se proponen mejoras a la app recién programada	Repaso de conceptos de variables, condiciones y ciclos. Uso de numero aleatorio, recuperación de un valor y comparaciones. Uso del elemento reloj.
Clase 2	Actividad 3: Bola 8 mágica (1ra versión)	Diseño y programación preliminar de la app, con sólo los elementos mas simples de la app. Se hace uso de un botón, un mensaje y un sonido.	Acercamiento al desarrollo de software iterativo. Concepto implícito de MVP
	Actividad 3: Bola 8 mágica (2da versión)	Arreglo de la disposición en pantalla de los elementos. Se agregan distintos mensajes que puede mostrar aleatoriamente.	Uso de listas
	Actividad 3: Bola 8 mágica (3ra versión)	Reemplazo del botón por un acelerómetro y adición un componente texto a voz.	Conceptos de requisitos.
	Actividad 4: El barco pirata (1ra versión)	Diseño de la vista y el controlador del deslizado del barco y el rebote contra los bordes.	Uso del elemento Canvas y SpriteImage
	Actividad 4: El barco pirata (2da versión)	Adición de una moneda que aparece aleatoriamente.	Uso de elementos de posición y repaso de valor aleatorio.
	Actividad 4: El barco pirata (3ra versión)	Detección de colisión del barco con la moneda y desaparición.	Condiciones iniciales y elementos visibles.
	Actividad 4: El barco pirata (4ta versión)	Múltiples monedas, por tiempo limitado. Propuestas de mejora al final.	Uso de reloj, duplicación de elementos, uso de bloque "SI...ENTONCES"
Clase 3	Diseño del proyecto	Diseñan el proyecto en torno a escenas, objetos, comportamientos y transiciones y nombres representativos. Cada objeto tiene una escena, descripción, función y modo de operar.	Enseñar la importancia de planificar antes de programar.
	Programar el proyecto	Programación de los elementos que definieron en la etapa anterior.	Uso de App Inventor
	Autoevaluación del avance	Evaluación del avance con respecto al plan, siguiendo la metodología de Keep-Fix-Try de manera implícita. Integración en el plan de cambios que surgen a lo largo del desarrollo.	Desarrollo de pensamiento critico al programar.

Clase 4	Programar mejoras del proyecto	Programación de los elementos que necesitan corrección.	Uso de App Inventor
	Autoevaluación del avance	Idem que el día anterior.	Desarrollo de pensamiento critico al programar.
Clase 5	Presentación de la app desarrollada	Se presenta a los compañeros, tutores y apoderados la aplicación desarrollada, explicando cómo funciona y qué hace.	Desarrollo de la habilidad de síntesis sobre el trabajo desarrollado.

En el contexto de este taller, se optó por usar App Inventor y celulares como entorno de desarrollo y herramientas, respectivamente, por la facilidad de acceso por parte de los niños y niñas. App Inventor es un entorno de programación visual para programación en bloques desarrollado por Google Labs[2]. Un lenguaje de programación en bloques se caracteriza por encapsular varias instrucciones en una estructura visual llamada bloque, que provee guía visual sobre su uso. La particularidad de App Inventor con respecto a otros lenguajes de programación por bloque es que los programas que se pueden producir son ejecutables en un aparato móvil, como un celular o una *tablet*. Otra característica de App Inventor es que está orientado a eventos, brindando una gran cantidad de bloques a la captura de distintos gestos y reacciones que se puedan generar de la interacción del usuario con los aparatos.

Una premisa detrás de la elección de App Inventor como herramienta de programación para el taller es, por un lado, la cercanía generacional de las niñas y niños al uso de aplicaciones móviles, *smartphones* y *tablets*. Además, se busca romper la idea de que la computación y la programación es algo relacionado sólo con los computadores, al ver cómo se puede programar un código y ejecutarlo en el dispositivo móvil.

3.2. Contraste con el trabajo relacionado

En vista del trabajo relacionado expuesto en el capítulo anterior, se seleccionan los aspectos a mantener de la versión original del taller. Primero, se considera adecuado mantener App Inventor como herramienta del taller, dado los argumentos expuestos a favor de su uso por parte de programadores novatos. Se diseñan actividades en App Inventor para acompañar los tópicos expuestos en cada sección.

Lo siguiente es que, dado que se decide seguir un diseño basado en trayectorias de aprendizaje para contribuir a las últimas propuestas, se debe revisar si lo que compone actualmente al taller contribuye a ello en alguna medida. Durante el análisis se identifica la intención de enseñar incrementalmente a programar, partiendo desde conceptos simples para el contexto (captura de un evento y reacción a él en la Actividad 1), aumentando en complejidad a medida que se avanza en el curso (manejo de varios eventos en la Actividad 4). Sin embargo, dado que el foco es sobre programar, no se logra observar que alguna actividad pueda ser reutilizada para darle un enfoque centrado en alguna trayectoria de aprendizaje de las propuestas por Rich et al.[17]. Por lo anterior, no se hace uso del material original.

Dado que es un taller remoto, la idea de tener un *handout* físico (como en la versión original), no

[2] `appinventor.mit.edu/`

es fácilmente replicable. Se decide mantener una guía para las actividades, pero en formato digital.

Con respecto a la propuesta de diseñar y programar una aplicación, se mantiene con algunos cambios, centrados en el hecho de que App Inventor maneja principalmente eventos.

3.3. Diseño

La solución que se propone es el rediseño del taller, tanto en el ámbito de estructura del plan como en el formato del curso. Esta solución fue evaluada usando la metodología de caso de estudio [23], con fuentes de datos tanto cualitativos como cuantitativos para analizar de manera global la recepción de las actividades basadas en trayectorias de aprendizaje. Se hace uso de esta metodología pues es un primer acercamiento a una mejora del taller, que consta de muchos factores disímiles con el anterior, y donde se quiere registrar el comportamiento de los involucrados.

En el caso de estudio se observó el comportamiento y desempeño de las niñas y niños que asistan al taller . Estos parámetros fueron medidos con diversos instrumentos: cuestionarios de entrada y salida, grabación de las sesiones y entrevistas de retroalimentación por parte del cuerpo docente. Luego los resultados fueron analizados a la luz de los distintos *papers* relacionados , para ver qué comportamientos y reacciones son esperables y replantear aquellas que se alejan de lo observado con anterioridad en experiencias comparables.

El taller, sus actividades y materiales, están rediseñados sobre los lineamientos planteados en el *paper "K-8 Learning Trajectories Derived from Research Literature: Sequence, Repetition, Conditionals"* [17], donde se propone un acercamiento a la enseñanza de pensamiento computacional en base a trayectorias de aprendizaje. En ese trabajo se identifican tres trayectorias fundamentales y necesarias para entender y desarrollar el pensamiento computacional en niñas y niños cursando octavo básico o menos: secuencias (Figura 3.1), condicionales (Figura 3.2) y repetición (Figura 3.3). Las figuras son traducciones al español hechas para este trabajo de memoria. En cada figura, se encuentran un conjunto de nociones, agrupadas según el nivel de conocimiento del alumno (principiante, intermedio y avanzado). Los cuadros grises indican nociones generales del pensamiento computacional y los cuadros blancos indican nociones aplicables a la programación. Las flechas indican que el bloque del cual sale es necesario para comprender el bloque al cual apunta. Si la flecha es negra, indica que es fundamental, mientras que si su color es gris, indica que facilita el entendimiento, pero no es imprescindible. También plantean un flujo incremental en la enseñanza de las nociones que son parte de cada trayectoria. Encerrado en colores están las nociones con las cuales se decidió trabajar en el taller.

Lo anterior es relevante por el área en la que se desarrolla : el Grupo de Interés Especial sobre la Enseñanza en Ciencias de la Computación (SIGCSE, por sus siglas en inglés) tiene 50 años [3]; la fundación del grupo de Innovación y Tecnología en la Educación de Ciencias de la Computación (ITICSE por sus siglas en inglés) data del año 1996 [4]; el concepto de pensamiento computacional, su caracterización e importancia de su enseñanza fue propuesto en el año 2006 por Jeannette Wing [22]. En este *paper* se menciona por primera vez el concepto de Pensamiento Computacional, describiéndolo como una habilidad tan importante como la lectura, la escritura y la aritmética. Caracteriza el pensamiento computacional de la siguiente manera:

[3] `https://dl.acm.org/doi/fullHtml/10.1145/3230687`
[4] `https://sigcse.org/events/iticse/index.html`

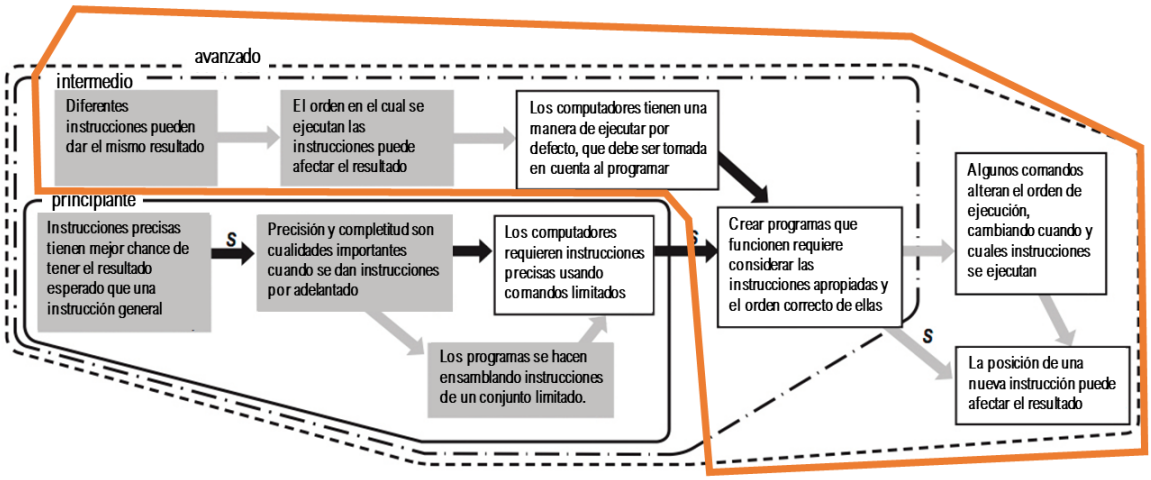

Figura 3.1: Trayectoria de aprendizaje para secuencias. Traducido al español. Original tomado de *K-8 Learning Trajectories Derived from Research Literature: Sequence, Repetition, Conditionals* [17]

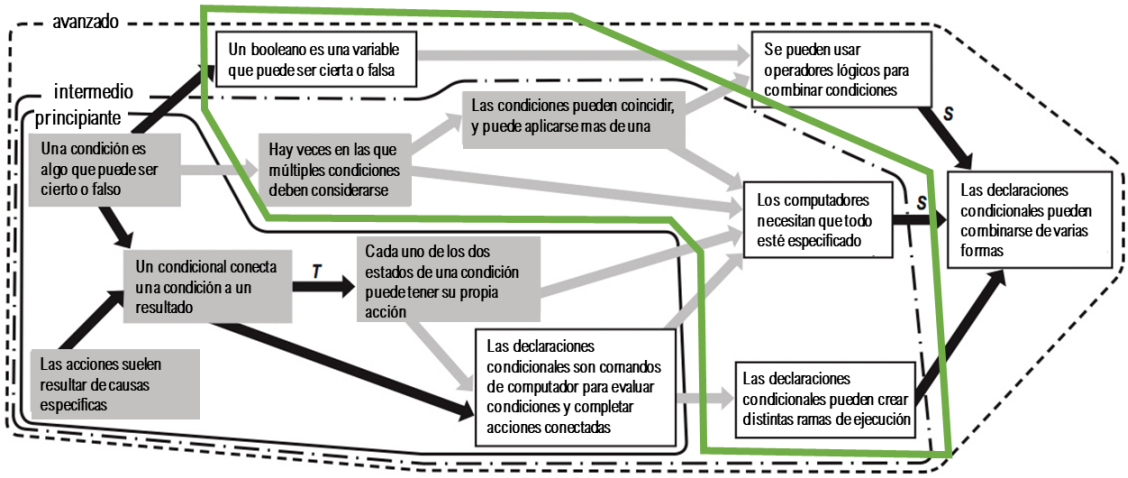

Figura 3.2: Trayectoria de aprendizaje para condicionales. Traducido al español. Original tomado de *K-8 Learning Trajectories Derived from Research Literature: Sequence, Repetition, Conditionals* [17]

- Conceptualizar, no programar
- Una habilidad fundamental, no una rutina mecánica
- Una forma en que los humanos, no los computadores, piensan
- Complementa y combina el pensamiento matemático e ingenieril
- Ideas, no artefactos
- Para todos, en cualquier parte

Si bien la propuesta hace mucho sentido , no es hasta el año 2017 cuando, mediante el estudio de currículos que promueven el pensamiento computacional, se proponen trayectorias de aprendizaje

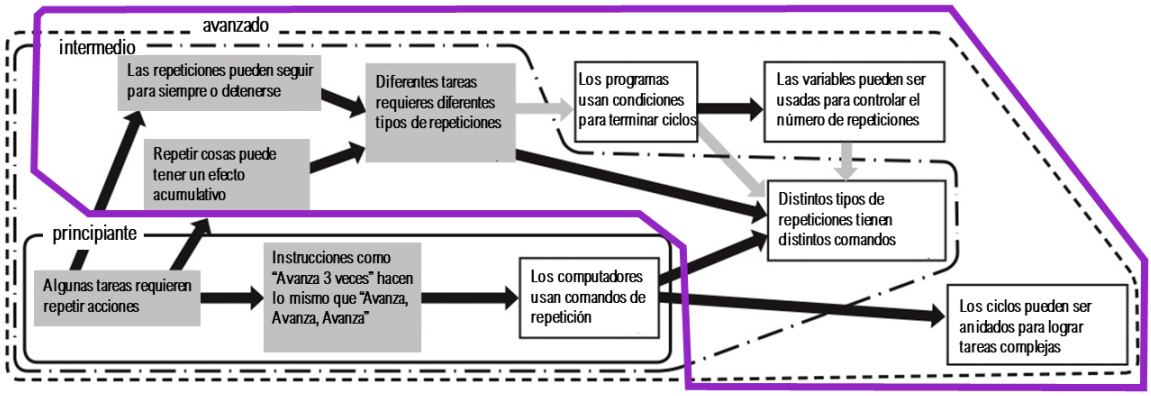

Figura 3.3: Trayectoria de aprendizaje para repeticiones. Traducido al español. Original tomado de *K-8 Learning Trajectories Derived from Research Literature: Sequence, Repetition, Conditionals* [17]

para desarrollar el pensamiento computacional [17]. En vista de lo reciente del trabajo que da soporte a esta memoria, es pertinente la idea de poner en práctica las propuestas de sus *papers*.

Dada la imposibilidad de ejecutar el taller de manera física por las restricciones de aislamiento social impuestas por la pandemia de COVID-19, la intervención se desarrolla en una plataforma virtual. De esta manera, se suple el espacio físico donde se ejecutaban las actividades anteriormente, manteniendo el foco de facilitar la interacción entre los distintos actores del taller.

El material del que se dispone es una guía, *snippets* de código y actividades, y se rediseñaron en dos etapas: primero, adaptando el currículum y creando actividades para que se ajustasen a la planificación de 5 días que tenía el taller original, con foco en las trayectorias de aprendizajes descritas en las figuras 3.1, 3.3 y 3.2. Tanto el material del curso original como la última versión se encuentran disponibles en `https://bit.ly/3AX97YI`. Luego de la primera modificación , se ajustó lo anterior a una planificación de 4 sábados (condición necesaria para compatibilizar el horario del taller con la disponibilidad de los niños y niñas), esta vez enfocado a la modalidad remota del taller.

3.3.1. Primera iteración

En la primera iteración se tomaron decisiones de diseño sobre el foco de las actividades y del taller en general. Se mantiene la estructura de 5 días y se mantiene el supuesto de clases presenciales. Primero, el taller se estructura en torno a una actividad central incremental: una aplicación de producción propia de las niñas y niños. Se busca motivar el aprendizaje de pensamiento computacional mediante involucrar a los asistentes en la creación de un artefacto computacional diseñado por ellos, que funciona gracias a lo que han aprendido en el curso [2]. También se persigue acercar a los asistentes al desarrollo de software de manera creativa y estructurada, guiándoles metodológicamente en el desarrollo de su proyecto.

Esta actividad fue trabajada en etapas de desarrollo incremental día a día: lluvia de ideas, convergencia, diseño y programación de la aplicación. De esta manera se encauzan las ideas de los asistentes y se vuelve alcanzable el objetivo de programar una aplicación en un tiempo acotado. El primer día se les asignan a los asistentes dos elementos de la paleta de App Inventor (medios, sensores o interfaces) y se motiva a las niñas y niños para proponer ideas de aplicaciones que haga

uso de ellas. El segundo día tiene como objetivo converger, decidiendo por una o una amalgama de las ideas propuestas. El tercer día tiene como objetivo planificar un diseño a nivel de vista y funcionalidades, y los días restantes se dedican a la programación de lo planificado. Lo anterior es un acercamiento inicial y guiado al desarrollo de software y sus lógicas internas, donde se toman los conocimientos adquiridos de la investigación de talleres anteriores de Scratch, que probaron que se puede enseñar nociones de desarrollo de software a niños y niñas si se hace de manera guiada y con lenguaje adecuado [11] . En la tabla 3.2 se describen las actividades a seguir (identificadas con un nombre y un código), su descripción y el objetivo que persigue.

Tabla 3.2: Actividades para el desarrollo del proyecto personal para el taller

Nombre de la actividad	Descripción	Objetivos
(P-1) Proyecto: Avance 1	Breve introducción a los distintos sensores y multimedia que pueden usar. Se les asigna al azar 2 elementos que deben incorporar en su app y desarrollar sus ideas desde ahí.	Profundizar el conocimiento de App Inventor, avivar su creatividad e introducirlos al desarrollo de software.
(P-2) Proyecto: Avance 2	Definir cuál de las ideas generadas en la clase anterior será programada.	Continuar con el trabajo de desarrollo de software.
(P-3) Proyecto: Avance 3	Definir las vistas de la aplicación, los objetivos de cada vista y repartir el trabajo de programación entre los participantes.	Nociones de MVC, continuar con el trabajo de desarrollo de software.
(P-4) Proyecto: Avance 4	Se les ayuda con la implementación, corrección de bugs y pruebas de uso con compañeros mediante actuación. Se les pregunta sobre qué les costó, qué ocuparon, y que describan su código y app.	Finalizar el ciclo de desarrollo de software.

Para aumentar las herramientas con las que cuentan las niñas y niños para su proyecto personal ,se antecede con actividades de menor duración que refuerzan las ideas base de las trayectorias de aprendizaje. En las figuras 3.1, 3.2 y 3.3 se muestran dentro de un lazo de color los conceptos escogidos para trabajar. Se seleccionaron los conceptos de rango medio y avanzado porque se espera que los conceptos más básicos los manejen por tener contacto desde temprana edad con elementos tecnológicos y por currículo escolar.

Las actividades se clasifican en dos grupos: con o sin programación. Las actividades que requieren programar usan *apps* desarrolladas en App Inventor. Las *apps* y los ejemplos fueron diseñados guiándose por los principios que plantea Jeannete Wing como definitorios del pensamiento computacional, presentados en subsección *Diseño* de este capítulo. Los insumos ocupados en las actividades se encuentran en el Anexo A. Se decidió que no todas las actividades fuesen de programación en pos de la duración de cada sesión del taller. En la Figura 3.4 se muestra cómo luce una *app* provista a los alumnos.

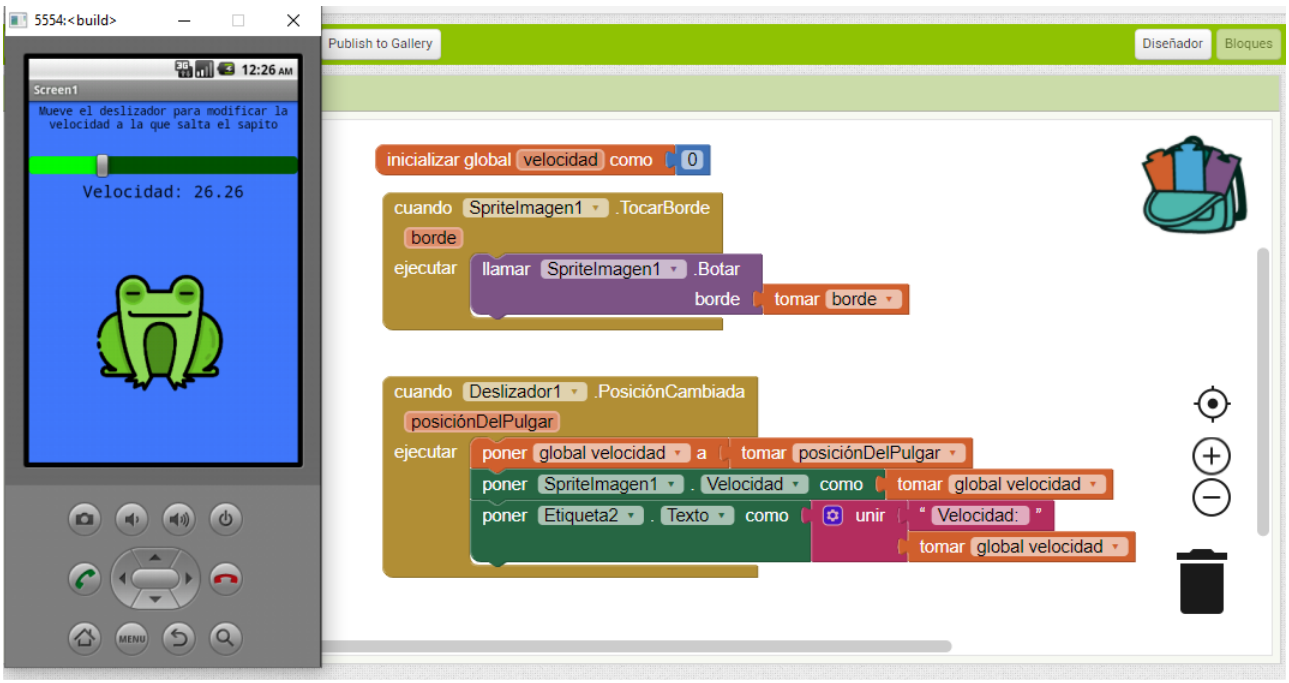

Figura 3.4: Captura de la sección de bloques de App Inventor y el emulador ejecutando el código

Por otro lado, las actividades que no necesitan programación hacen uso de ejemplos y reflexiones por parte de los niños en torno a preguntas planteadas por el docente a cargo de la sesión. Estas preguntas tienen como propósito que conecten el pensamiento computacional con situaciones cotidianas, en las cuales pueden o no interactuar con tecnología. Así, se busca lograr que lo incorporen como una habilidad más, que puede ser después usada específicamente para programar. El detalle de las preguntas se encuentra en Anexo A.

Cada una de estas actividades se clasifica por la trayectoria de aprendizaje que refuerza: secuencia, condicionales y repetición. Las actividades se presentan en la Tabla 3.3, la Tabla 3.4 y la Tabla 3.5 . Del mismo modo que en la tabla de actividades anterior, cada elemento tiene un identificador único, descripción de la actividad y los objetivos a los que apunta, donde el color naranjo corresponde a Secuencia, el color verde a Condicionales y el color morado a Repeticiones. Los artefactos relevantes se encuentran en el Anexo A.

Tabla 3.3: Actividades para la trayectoria de aprendizaje de secuencias para el taller

Nombre de la actividad	Descripción	Objetivos
(S-1) Todos los caminos llevan a Roma	Primero, se explica el concepto con situaciones cotidianas (atarse los zapatos, dibujar un árbol, contar huevos). Luego, se muestran ecuaciones matemáticas que producen el mismo resultado y oraciones que describen la misma situación.	Relacionar el concepto "diferentes instrucciones pueden dar el mismo resultado.[en] su entorno.

(S-2) No es lo mismo gato que toga	Se presentan la idea de que las recetas de cocinas deben ser hechas en cierto orden, y se les cuenta que lo mismo sucede con la programación.	Relacionar el concepto .[el] orden en el cual se ejecutan las instrucciones afecta el resultadoçon experiencias propias y entender que al programar es lo mismo.
(S-3) El computador tonto: La revancha	Se les habla sobre el orden que sigue el computador al leer un código. Se les muestra distintos bloques de pseudocódigo y deben discutir y concluir sobre cuál será el resultado de las secuencias mostradas.	Presentación del concepto “los computadores tienen una manera de ejecutar por defecto, que debe ser tomada en cuenta al programar".
(S-4) Cada cosa en su lugar, y lo que no sirve se guarda	Recordar lo visto en las clases pasadas, con énfasis en que deben ser ordenados al programar. Se les provee una app con bloques extras de los necesarios para el correcto funcionamiento según la descripción.	Explicación del concepto “programar requiere considerar las instrucciones apropiadas y el orden correcto de ellasz uso de App Inventor como vehículo para el concepto.
(S-5) Distintas elecciones, distintas historias	Se les piden ejemplos donde hay distintas “rutas". Se introducen los conceptos de IF y ELSE y ejemplos de uso. Se les entrega una app con el juego Pac-Man que no tiene controles y que deben arreglar.	Presentación del concepto “Algunos comandos alteran el orden de ejecución, cambiando cuándo y cuáles instrucciones se ejecutan". Uso de App Inventor.
(S-6) Revuelto como huevos	Se les muestra cómo el concepto está presente en muchas acciones que ya realizan (rutinas diarias, cocinar). Se les pide que agreguen una nueva instancia a la aplicación que trabajaron en la actividad “Yo tenía diez perritos...".	Explicación del concepto “la posición de una nueva instrucción puede afectar el resultado". Ejercitación del concepto mediante App Inventor.

Tabla 3.4: Actividades para la trayectoria de aprendizaje de condicionales para el taller

Nombre de la actividad	Descripción	Objetivos
(C-1) El computador tonto	Se les explica que los computadores sólo saben seguir instrucciones y que éstas deben estar bien explicadas. Luego deben arreglar un código que no funciona como se describe que debe hacerlo.	Internar el concepto "los computadores necesitan que todo esté especificado" mediante la explicación y la experiencia.

(C-2) Que si, que no, que nunca te decides	Se explicará el concepto de verdadero y falso en el contexto computacional. Se mostrará su uso como input y output de distintas operaciones cotidianas (“s“i hace frío, salgo abrigado", “revisa si es antes de las 12"). Se hace uso de una app para aplicar el concepto.	Explicación del concepto “un booleano es una condición que puede ser cierta o falsaz la exploración del concepto usando App Inventor.
(C-3) Detrás de una acción, están sus condiciones	Se les muestra cómo una acción cotidiana (de elección del curso) se compone de varias condiciones que deben ser revisadas previamente. Se les explica también que describir el estado de un objeto requiere múltiples condiciones.	Relación del concepto “múltiples condiciones deben ser consideradas" con “las condiciones pueden coincidir y también puede aplicarse más de una".
(C-4) Final bueno, final malo, final marciano	Se relacionan conceptos cotidianos con la programación de “rutas"que se discutió en la clase anterior. Se les provee una app para que experimenten con el concepto de que distintas opciones cambian el resultado.	Explicación del concepto "las declaraciones condicionales pueden crear distintas ramas de ejecución", junto con el uso de App Inventor como usuario.

Tabla 3.5: Actividades para la trayectoria de aprendizaje de repeticiones para el taller

Nombre de la actividad	Descripción	Objetivos
(R-1) No pares, sigue, ¡sigue!	Explicación de distintas repeticiones con ejemplos cotidianos (repetición de las rutinas, de la salida y puesta del sol, en contraste con la lluvia y la cocina). Se les pide que den ejemplos de cada tipo de repetición y que piensen qué pasaría si se invirtiesen los tipos de repeticiones.	Relacionar el concepto “las repeticiones pueden seguir para siempre o detenerse.[en] su entorno.
(R-2) Una y otra vez y¡quedará todo limpio!	Se les explica el concepto mediante ejemplos que usan acumulación: masticar, barrer, caminar. Se pedirán más ejemplos parecidos. Deberán corregir una app donde la repetición no genera acumulación, pero debería hacerlo.	Explicación del concepto “repetir cosas puede tener un efecto acumulativoçon ejemplos cotidianos y la exploración del concepto usando App Inventor.

(R-3) La herramienta adecuada para cada tarea	Haciendo referencia a las repeticiones hasta cierto punto y las que son para siempre, se pide que clasifiquen casos de la vida real como uno u otro y que justifiquen las respuestas. Se introducirán los conceptos de WHILE y FOR, sus diferencias y usos.	Relación del concepto “diferentes tipos de repetición tienen diferentes comandos" con "diferentes tareas requieres diferentes tipos de repeticiones".
(R-4) Si sale el sol, saldremos a jugar	Se nombraran ejemplos de la vida cotidiana donde ciertas condiciones significan cambiar de acción (“cuando pasa algo, entonces sucede esto otro"). Se les pedirá que den ejemplos acorde al caso.	Introducir el concepto de “los programas usan condiciones para terminar ciclos".
(R-5) Yo tenía diez perritos...	Se ilustra el concepto con repeticiones cotidianas que dependen de números o cantidades. Luego se les provee una app para que experimenten con cambiar una variable y observar cambios.	Presentación del concepto “las variables pueden ser usadas para controlar el número de repeticiones". Observación de una app para cambiarla o mejorarla.
(R-6) Igual que las Matrioshka	Se les presenta el concepto mediante asociaciones cotidianas y de ciencias naturales (ciclos naturales, órbitas de los planetas).	Presentación del concepto “los ciclos pueden ser anidados para lograr tareas complejas".

Finalmente, se diseñan actividades adicionales que ayudan al desarrollo de la clase en distintos aspectos. En particular, se mantiene durante la primera iteración la introducción a App Inventor como se hacía en la versión original del curso. La Tabla 3.6 presenta la información sobre las actividades y sus objetivos.

Tabla 3.6: Actividades adicionales para el taller

Nombre de la actividad	Descripción	Objetivos
(A-1) Introducción a App Inventor	Explicación de qué es App Inventor y cómo empezar a usarlo. Se compara con Scratch.	Familiarizarlos con el uso de la herramienta.
(A-2) Kahoot!	Cuestionario simple con preguntas basadas en (S-1), (S-2), (C-1) y (R-1).	Rompe-hielo y refresco de memoria
(A-3) Kahoot!	Cuestionario simple con preguntas basadas en (S-3), (S-4), (C-2) y (R-2)	Rompe-hielo y refresco de memoria
(A-4) Kahoot!	Cuestionario simple con preguntas basadas en (S-5), (C-3), (R-3) y (R-4)	Rompe-hielo y refresco de memoria

Como resultado de la primera iteración de rediseño del taller, se tiene una planificación inicial

con actividades y material de apoyo. Siguiendo el diseño del taller original, está planeado para que cada clase se pueda desarrollar en un espacio de 2 horas y media o 3 horas, en caso de ser necesario. La planificación se observa en la tabla 3.7, donde cada día tiene el código de las actividades que se deben desarrollar, con el color de la trayectoria de aprendizaje asociada.

Tabla 3.7: Primera planificación para el taller

<table>
<tr><td>Día 1</td><td>Día 2</td><td>Día 3</td><td>Día 4</td><td>Día 5</td></tr>
<tr><td>S-1</td><td>S-2</td><td>S-4</td><td>S-5</td><td>S-6</td></tr>
<tr><td>R-1</td><td>S-3</td><td>R-3</td><td>R-4</td><td>R-6</td></tr>
<tr><td colspan="5">Recreo</td></tr>
<tr><td>C-1</td><td>R-2</td><td>C-3</td><td>R-5</td><td rowspan="3">P-4</td></tr>
<tr><td>A-1</td><td>C-2</td><td rowspan="2">P-3</td><td>C-4</td></tr>
<tr><td>P-1</td><td>P-2</td><td>P-4</td></tr>
</table>

Una vez terminada la primera iteración, se validó con un experto del dominio, con experiencia en el diseño y planificación de intervenciones para fomentar el desarrollo del pensamiento computacional en K-8 que el objetivo de presentar una primera planificación en torno a trayectorias de aprendizaje se cumplió. Dado que cada actividad es medianamente independiente de las otras (con la única restricción de que ciertos contenidos deben ser presentados antes que otros, siguiendo lo descrito en las Figuras 3.1, Figura 3.2 y Figura 3.3), el ajuste a una cantidad de días distintos es relativamente fácil.

3.3.2. Segunda iteración

En la segunda iteración, se ajustaron las actividades a un esquema de 4 días , los cuales no son consecutivos. Esta decisión viene dada por el contexto escolar chileno actual. En Chile la Jornada Escolar Completa es el modo de dictar clases con mayor presencia, la cual contempla la asistencia de las niñas y niños a clases dictadas por su establecimiento educacional durante 8 horas diarias[5]. Esto significa cansancio y menos foco al final del día. En vista de lo anterior, se evaluó que dictar el taller en una modalidad extra-programática tendría peor recepción que una modalidad de clases los sábados por la mañana.

En vista de esto, se decidió agregar un tipo nuevo de actividad para refrescar la memoria de los asistentes y ayudarles a conectar con el contenido de la clase actual. Adicionalmente, y teniendo en cuenta que el foco para esta segunda iteración es la característica remota de las clases, se agregaron descansos de 15 minutos entre bloques. Esto para cambiar el foco a los asistentes y al equipo docente y evitar la fatiga de los participantes.

Se reformuló el material para ajustar las presentaciones a un esquema de cuatro clases. Además, las actividades *unplugged*, o desconectadas de la programación, que debían ser respondidas en un cuadernillo se modificaron para ser preguntas de respuestas abiertas, de discusión y reflexión para todos los asistentes. En la figura 3.5 se muestra un ejemplo de presentación de una actividad desconectada.

[5] `http://bcn.cl/2f8qo`

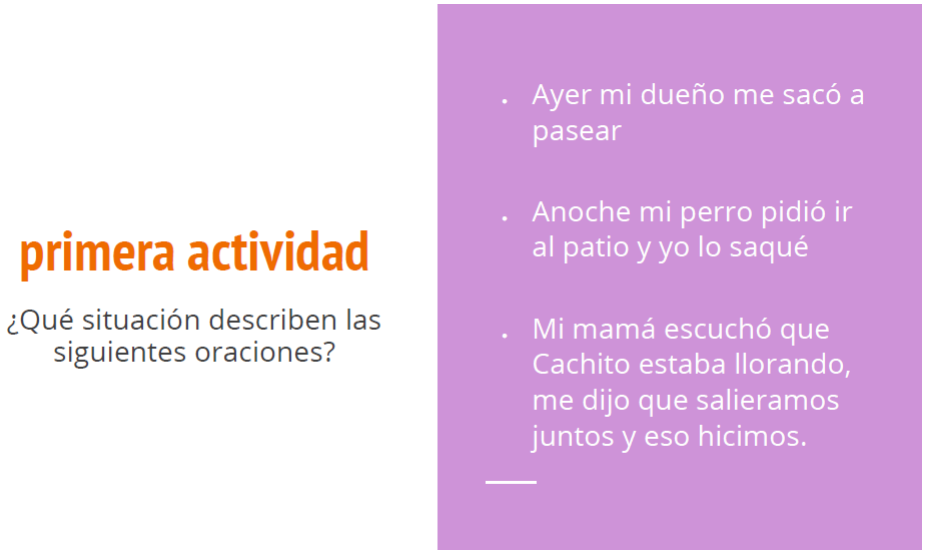

Figura 3.5: Captura de actividad *unplugged*

El resultado de la segunda iteración es una nueva planificación, esta vez para cuatro días. Se tiene también que el material fue modificado para ajustarse a este nuevo esquema temporal. Como en la iteración anterior, se presenta en la Tabla 3.8 el itinerario a seguir durante el taller. Se indica mediante los códigos qué actividades se realizarán en cada jornada.

Tabla 3.8: Segunda planificación para el taller

Día 1	Día 2	Día 3	Día 4
A-1	A-2	A-3	A-4
S-1	S-3	R-3	C-4
R-1	R-2	R-4	R-5
Recreo			
S-2	C-2	C-3	R-6
C-1	S-4	S-5	S-6
Recreo			
P-1	P-2	P-3	P-4

En síntesis, se cuenta con la definición del diseño instruccional del taller, incluyendo actividades de programación, plantillas de respuestas a las preguntas y *snippets* de código funcional. Esto está disponibilizado para los tutores mediante una carpeta compartida de Dropbox. Todo el material rediseñado, junto a las decisiones de estructura para la ejecución del taller fueron piloteados con los tutores del curso previo al taller, a fin de validar la infraestructura elegida, así como evaluar su estabilidad técnica. Luego de la evaluación y validación de la plataforma elegida (en este caso, Zoom), se ejecuta una instancia formal del taller con una muestra de estudiantes, recopilando datos durante el proceso. Finalmente, dichos datos se analizaron en torno al problema planteado y se propusieron sugerencias para futuras implementaciones. El proceso se describe en profundidad en el capítulo siguiente.

Capítulo 4

Ejecución del taller

En este capítulo se discute la puesta en marcha a nivel técnico tomadas en el capítulo anterior. Además, se presentan las elecciones de implementación tomadas en última instancia y las lógicas detrás de ellas.

La solución que se implementa en esta memoria es un taller remoto basado en App Invento para promover el desarrollo del Pensamiento Computacional, cuyo diseño está guiado por las trayectorias de aprendizaje propuestas por [17]. El taller se instancia de manera remota haciendo uso de la plataforma Zoom.

4.1. Preparación para la ejecución del taller

Primero, el taller se ofrece para el rango de quinto a séptimo básico para motivarles desde temprana edad el interés por la ciencia, tecnología, ingeniería y matemáticas (STEM por sus siglas en inglés), área a la cual pertenece la computación . Esto se hace mediante un formulario de inscripción que se hace circular entre integrantes de la comunidad del Departamento de Ciencias de la Computación y de la Facultad de Ciencias Físicas y Matemáticas. Se disponibiliza a gente cercana a una comunidad universitaria como muestreo por conveniencia. De esta manera, se busca reducir la reticencia a participar en esta instancia. Se espera que, dado el contexto de los participantes, haya un sesgo hacia el contacto temprano con la tecnología. Sumado al formulario de inscripción, se adjunta una lista de requerimientos básicos de los cuales deben disponer los alumnos para poder participar adecuadamente en el taller. Esto incluye acceso a internet, cuenta de Google para acceder a la plataforma de App Inventor, un navegador en el computador, entre otros. Tanto el formulario (Anexo D) como la lista de requerimientos (Anexo A) se encuentran en los Anexos.

Una vez se cierran las inscripciones, se les envían dos correos a los apoderados: uno con un formulario de consentimiento informado y de asentimiento para los participantes, explicando de qué se trata el taller, qué cosas se harán y en qué ámbitos se necesita su autorización. Esto es porque el estudio fue aprobado por el Comité de Ética de la Facultad, dado que se trabajarán con personas ajenas a la institución y menores de edad. El otro correo (Anexo C) contiene un manual de instalación de App Inventor, que explica en español los pasos a seguir para elegir la modalidad adecuada e instalarla sin asistencia, en preparación al taller. El manual (Anexo A) brindado a los

participantes espera reducir en esta instancia la brecha presente en los lenguajes de programación, que casi en su totalidad están en inglés, lo que lo hace inaccesible a los hispanohablantes nativos [20].

Con respecto a la cantidad de docentes por alumnos, se tiene el conocimiento experiencial de participaciones anteriores en el Taller de Scratch, brindado por la misma Universidad, que una proporción de un docente por cada 5 alumnos es buena para trabajar presencialmente [12]. Se apuntó a mantener una proporción similar en vista de esa experiencia. Dado que se inscribieron 29 participantes, es necesario el apoyo de más personas que sólo el profesor que dicta la clase. Por ello, se hace un llamado abierto a participar como ayudantes de la actividad en los canales destinados a alumnos del Departamento de Ciencias de Computación. Se les solicita que tengan interés en enseñar e idealmente experiencia trabajando con niñas y niños. Se reúnen 3 personas más, a las cuales se les informa el objetivo del experimento, las responsabilidades y tareas que deberán desarrollar. Se les da una introducción a App Inventor y, previo a cada clase, un repaso por Zoom del contenido a revisar y las actividades que se realizarán, en qué modalidad, una manera de resolverlos y cuáles son los resultados esperados. Al finalizar cada sesión del taller, se hace un repaso por lo que se hizo en el día, se recogen las impresiones sobre el desempeño de los alumnos y se lleva a cabo un Keep-Fix-Try informal a fin de mejorar la siguiente sesión. Esto genera insumos para responder las preguntas de validación del caso de estudio.

Tanto a la entrada como a la salida del taller se les entrega un cuestionario a los asistentes (el cual se encuentra en el Anexo D) para recoger sus impresiones previas a su paso por el taller y posteriores a él. Los cuestionarios usados son una adaptación a Google Form [1], (una plataforma para generar encuestas de manera digital) de los cuestionarios usados en implementaciones anteriores del Taller de Scratch [11], brindado por la misma Universidad y Departamento. Cuenta con preguntas sobre su conocimiento de computación y su experiencia usando aparatos electrónicos, además de los conceptos que relaciona a esta área. Algunas preguntas deben responderse de manera escrita y otras mediante dibujos. Distan de la implementación original al ser un cuestionario digital y no físico, modificación necesaria para poder distribuir la herramienta de medición en una instancia remota. A diferencia de la instancia presencial, donde se distribuían las encuestas a cada asistente en papel, las encuestas fueron enviadas por correo a los apoderados, con la instrucción de que fuesen contestadas por sus pupilos.

Se decidió hacer uso de la plataforma Zoom [2] para dictar el taller dado que, al momento de evaluar distintos servicios que brindasen la posibilidad de videollamadas, esta poseía la capacidad de generar *breakout rooms* (salas aparte de la sesión principal) donde se conservaba la capacidad de grabar la sesión. Las grabaciones son fundamentales para el análisis posterior del taller. Si bien otras plataformas también ofrecen la grabación de sesiones en salas aparte, Zoom cuenta con un factor de familiaridad (por su uso en colegios en el contexto de clases remotas) que hizo que el uso de la herramienta fuese mucho más fluido por el lado de los participantes.

Se hizo uso de la plataforma Kahoot [3] como vehículo de algunas las actividades adicionales porque facilita generar cuestionarios con un enfoque lúdico, y se ha visto en otros talleres que su uso potencia la participación en clases para esta generación de alumnos [13]. Vale la pena mencionar también que era previamente conocido por el cuerpo docente, por lo que la capacitación al cuerpo docente sobre esta herramienta no tuvo mayores contratiempos. Los cuestionarios se encuentran en el Anexo E, Anexo F y Anexo G.

[1] `https://www.google.com/intl/es-419_cl/forms/about/`
[2] `https://explore.zoom.us/en/about/`
[3] `https://kahoot.com/company/`

En vista de los anterior, el taller se dictó mediante la plataforma Zoom, donde se hizo uso de la capacidad de dividir a los participantes en sub-grupos (breakout rooms) y la grabación de las clases para posterior análisis, todo esto con el previo consentimiento informado a los apoderados y asentimiento de los participantes. El chat por escrito también fue relevante, pues brindó a los tutores la capacidad de compartir enlaces a todos los participantes y a los asistentes les dejó mantener conversaciones en paralelo a la clase, instancia muy importante para socializar. Se hizo uso también de Kahoot para facilitar las actividades rompehielo al inicio de cada clase, las cuales consistían en un repaso superficial de los tópicos tratados en la sesión de la semana anterior. Dado que la herramienta elegida para programar en esta instancia es App Inventor, también se reporta (mediante observación y toma de notas) el uso de su plataforma web [4], aplicaciones para celular y emuladores por parte de los tutores y los alumnos. Lo anterior a fin de entender cómo se relacionaron con esta herramienta y si se puede mejorar algún aspecto de ello en el ámbito del taller.

4.2. Ejecución del taller

El taller se dictó durante los sábados en la mañana desde el 29 de Mayo al 19 de Junio del año 2021, para facilitar que los asistentes participen activamente en clases, por lo expuesto en el capítulo anterior. La duración de la sesión en total y de cada bloque viene dada por experiencias compartidas en entrevistas a profesores que han dictado clases a niñas y niños de quinto a séptimo básico en periodo de distanciamiento social. Ellos han observado que la retención de interés es menor que en clases presenciales [15], lo que se condice con la encuesta aplicada a los . Adicionalmente reportan que los recreos cobran suma importancia en esta nueva modalidad, pues son instancias de descanso y recuperación en miras al próximo bloque, tanto para los alumnos como para el cuerpo docente.

Cada sesión dura 2 horas y media, dividida en 3 bloques: el primero y el segundo tienen 45 minutos de trabajo y 15 de descanso, mientras que el tercero es de 30 minutos de trabajo. A su vez, cada bloque de trabajo se divide en un periodo expositivo de largo variable y otro periodo para desarrollar actividades relacionadas a la materia recién vista, como lo indica la figura 3.8. El primer bloque cuenta con una actividad rompe-hielo (Kahoot) para que conecten con la dinámica de la clase y repasen de manera lúdica los contenidos de la clase anterior, en caso de haber. Luego, tanto el primero como el segundo bloque se usan para presentar conceptos de Pensamientos Computacional y desarrollar actividades en App Inventor que tuviesen relación con los conceptos presentados, con la guía y ayuda del cuerpo docente. Estas actividades tienen como objetivo proveer de mayor perspectiva y más herramientas a los niños y niñas que asistan al taller, en miras al desarrollo de su propia *app*. Finalmente, el tercer bloque está reservado para el trabajo del proyecto en que cual avanzarán las y los niños durante el taller. Cada vez que se deba trabajar en actividades, se crean *breakout rooms*, donde cada una estaba liderada por un miembro del equipo docente y cada niño (asignados al azar en la primera clase) va a la sala del tutor que le corresponde, para trabajar en grupos mas reducidos y así tener una atención más personalizada de sus dudas.

El material[5], tanto las presentaciones como las actividades, es una adaptación del diseño original del curso de App Inventor piloteado el año 2018. Este contaba con una guía de trabajo, apoyo visual esporádico y actividades tanto desconectadas (donde los alumnos conversaban entre sí, respondían en la guía y/o respondían a mano alzada) como conectadas (donde trabajaban en el computador ejecutando código o programando). El motivo de esta reforma es tanto para adaptarse a las pro-

[4] `http://ai2.appinventor.mit.edu`
[5] `https://bit.ly/3AX97YI`

puestas metodológicas más recientes (lo cuál se explica en el capítulo anterior), como para facilitar la adquisición de conocimiento en un contexto remoto. En particular y haciendo caso a este último motivo, los cambios identificables son los siguientes:

- La guía de trabajo se transforma en una presentación de diapositivas. En esta presentación se introducen y trabajan los conceptos, como se muestra en la figura 4.1, donde se tiene el título de la actividad, el concepto que se les quiere transmitir e imágenes relacionadas con los ejemplos que se usan para transmitir el concepto. También se enuncian las actividades desconectadas usando la presentación, como en la figura 3.5, donde se plantea a la izquierda la pregunta y a la derecha las situaciones a evaluar. Además se da el contexto necesario e instrucciones para las actividades conectadas, como se muestra en la figura 4.2, que plantea el problema que se debe resolver. Se decidió no exigir la impresión de la guía de trabajo para evitar que la realización exitosa del taller dependiese de factores externos (como la disponibilidad de impresoras) y fuera del control del equipo docente.
- La mayoría de las actividades desconectadas se reformaron como actividades en App Inventor mediante la creación de actividades alternativas que persiguieran el mismo objetivo de aprendizaje, siendo validadas por los tutores mediante la discusión de si se logra obtener el aprendizaje esperado luego de desarrollarla. Dado que en un contexto remoto las interacciones ocurren con menor naturalidad [1], no se pueden llevar acabo aquellas que dependen de que los alumnos conversen entre si de manera espontánea, como era el caso del taller de App Inventor en su diseño original. Esto significa la pérdida de discusiones espontáneas y conexiones de conceptos que se podrían dar mediante el diálogo. En su lugar, se crearon actividades conectadas nuevas que potenciaban el mismo hito de aprendizaje. Se validó con los tutores mediante comentarios sobre la facilidad de resolución en vista de la materia vista anteriormente.

Figura 4.1: Captura de presentación de conceptos

Con respecto al código provisto para las actividades conectadas (el cual se validó con los tutores mediante comentarios sobre la facilidad de resolución en vista de la materia vista anteriormente) los programas siempre se presentan incompletos, pero con la motivación de investigarlos y arreglarlos. Se eligió presentar así el código por tres motivos: el primero es para romper con la idea de que

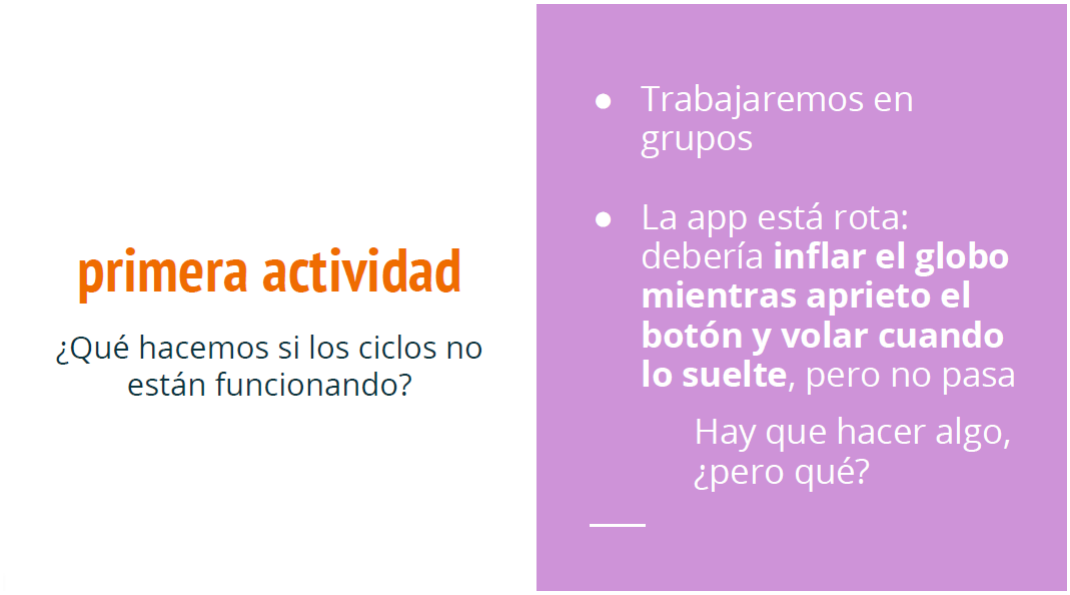

Figura 4.2: Captura de presentación de una actividad

todo lo relacionado a computadores funciona bien. El segundo es para involucrarlos en las lógicas de programación de una *app*, alejándoles de la experiencia común de ser sólo usuario, lo cual se logra bien con App Inventor , según lo investigado por K. Roy [18]. La tercera motivación es facilitar el espacio para tener la satisfacción de arreglar algo con los conocimientos que se tienen en ese momento, creando así un refuerzo positivo entre las actividades y su desarrollo. Se puede observar en la figura 4.2 que hay un vacío entre lo que se tiene (un programa que no registra ciertos eventos) y lo que se quiere lograr (que la *app* los registre y reaccione a ellos).

La decisión de dedicar parte del taller al desarrollo de un proyecto ideado por ellos, fomentando la creatividad con dinámicas de lluvia de ideas y empoderándolos sobre sus creaciones, guarda relación con que el aprendizaje es más efectivo si existe una componente creativa y personal en el proceso. Según A. Basawapatna, A. Repenning, M. Savignano, J. Manera, N. Escherle y L. Repenning, las personas se involucran más en aprender si intuyen que las herramientas que les son enseñadas pueden ser usadas en un proyecto personal [2].

El proyecto fue trabajado durante las cuatro sesiones de manera incremental y ordenada, guiado por los tutores. El objetivo del proyecto es desarrollar una *app*, lo cual se sabe que es una manera efectiva de consolidar cierto grado de conocimiento de pensamiento computacional, gracias a las investigaciones presentadas por Grover [10]. Se facilitó la lluvia de ideas para poder elegir cómo será su proyecto, con "¿qué aplicación podemos desarrollar que use los elementos que nos asignaron?" como la pregunta central. Se facilitó que tomaran acuerdos sobre el contenido y diseño del proyecto y se les ayudó a que dividieran responsabilidades de programación entre los integrantes de cada grupo. Al final, el tutor lleva a cabo la unificación de todo lo anteriormente programado por las niñas y niños.

Los ejemplos y referencias usados como vehículo para los conceptos de Pensamiento Computacional se buscó que cumplieran dos requisitos: ser relevantes para ellos como grupo etario y ser lo más cercano a experiencias cotidianas que tuviesen todos, definido en torno a experiencias esperables para su edad e intereses de moda entre sus pares (como juegos, redes sociales, entre otros). A nivel

de identidad gráfica del taller, se hizo uso de iconos de Flaticon [6] por su simplicidad, fácil obtención y coherencia visual.

A la hora de implementar el taller, se tomaron decisiones de diseño en vista de las dinámicas observadas. La más importante es que el proyecto en el cual avanzaron las y los niños se hizo en grupos y no de manera individual, al ver que se observó que se sentían mucho mas cómodos así. Se tiene el respaldo para este cambio en la investigación sobre proyectos libres de Shuchi Grover [10], donde se observa que los proyectos grupales logran mejores resultados a nivel de retención de conceptos que los proyectos individuales.

4.3. Análisis de la ejecución

La clasificación de la información y análisis del taller se lleva a cabo en tres frentes:

- Revisión de las grabaciones en video de cada clase (capturados haciendo uso de la plataforma Zoom), con posterior anotación en un tablero (Anexo D) de Google Jamboard de los conceptos y conductas relevantes. Estos se clasifican en los ámbitos operacionales (relacionado a la implementación de las herramientas), actitudinales (relacionado a la disposición que presentan en clases) y cognitivos (relacionado a los conocimientos que exhiben). Estas tres categorías son las que se identifican como importante en la implementación y el desarrollo del taller.
- Evaluación a nivel cuantitativo y cualitativo de las encuestas de entrada y salida contestadas por los asistentes, siguiendo las metodologías usadas en los talleres de Scratch que hacen uso de la misma herramienta.
- Revisión de las anotaciones posteriores a cada clase en las sesiones retrospectivas con el cuerpo docente, las cuales fueron discusiones con preguntas guiadas bajo el marco de Keep-Fix-Try.

Cada una de estas instancias aporta distintos puntos de vista valiosos para la comprensión de esta implementación, lo que se logra al responder las preguntas de validación planteadas en el marco de un caso de estudio. Adicionalmente, abre la posibilidad de mejorar el taller en miras de próximas versiones. La organización de los datos se realiza en Google Jamboard para visibilizar, organizar los conceptos relevantes que surgen y facilitar su análisis. Los archivos usados (video y código) se disponibilizan tanto en Dropbox como en Google Drive, los cuales son manejados sólo por el equipo docente y con fines educativos y de investigación.

[6] `https://www.flaticon.es`

Capítulo 5

Validación: Caso de Estudio Único

Para validar el taller realizado como solución al problema de la enseñanza remota de pensamiento computacional a niñas y niños, se analizó el taller siguiendo la metodología de un caso de estudio. Se hace uso del protocolo propuesto por Robert K. Yin [23] para llevar a cabo la validación a nivel técnico de la solución propuesta.

5.1. Visión general del proyecto de caso de estudio

Según Yin, la estrategia de caso de estudio es óptima para estudiar eventos contemporáneos contextualizados en un entorno real, no artificial ni controlado [23]. Como se busca monitorear el desarrollo de un curso para entender su recepción del curso, el comportamiento de los participantes está fuera de las variables que se puedan controlar y por lo mismo, es interesante observarlo. El hecho que el taller se dicte sobre un esquema de trayectorias de aprendizaje de manera remota es una combinación novedosa, por lo que vale la pena ver cómo es recibida por los asistentes y, por ende, se ajusta al criterio planteado.

5.2. Protocolo para caso de estudio

El contexto y la delimitación de la investigación son niños chilenos entre los 9 y 11 años de edad, relacionados con trabajadores de la Universidad de Chile, en contexto de distanciamiento social.

Teniendo en cuenta el contexto en el cual se desarrolla la memoria, surgen preguntas como ¿En qué medida afectará que el taller sea remoto?, ¿qué tanto incorporarán de los contenidos los alumnos?, ¿con qué actividades conectarán más los alumnos?, ¿qué dinámicas se darán entre las niñas y niños?, entre otras que se explicitan más adelante. Siguiendo la propuesta de Yin, estas preguntas se estructuran en 5 grupos, los cuales van desde los más especifico (preguntas que se realizan a participantes específicos) a los mas global (preguntas transversales a todos los casos en un estudio).

La hipótesis que se pretende evaluar es que, *si se siguen trayectorias de aprendizaje y se toman*

los resguardos necesarios para amortiguar el efecto de la distancia en clases remotas, es posible que niñas y niños de entre quinto a séptimo básico desarrollen competencias básicas de pensamiento computacional en un curso de 4 sesiones. Este nivel de competencia viene dado por las trayectorias de aprendizaje propuestas en el trabajo de K. Rich et al [17], donde el mínimo en este taller son las habilidades expuestas como nivel intermedio.

Un supuesto que se maneja para el análisis del taller es que los niños a esa edad no tienen herramientas de pensamiento computacional en su formación escolar, dado que el tema no se aborda explícitamente en las bases curriculares del MINEDUC. También y en el caso particular de este taller, se espera que esta experiencia desarrolle un interés por la computación en ellos. Se planea medir esto último con preguntas a los padres al final del taller, para medir el impacto a corto plazo.

Los antecedentes para este caso de estudio es el cuerpo bibliográfico previamente presentado y analizado en el capítulo 2. En particular, todo lo relacionado a la investigación que se está llevando actualmente sobre la formalización de la enseñanza de Pensamiento Computacional: cómo se conceptualiza el Pensamiento Computacional, cómo han sido hasta ahora los intentos de enseñanza y cuáles son las características de las últimas propuestas. En breve, si bien existe gran trabajo avanzado tanto en la educación de pensamiento computacional como de la enseñanza a distancia, la intersección entre ambos dominios es muy escasa. Adicionalmente, del área de la enseñanza de pensamiento computacional se propone el uso de trayectorias de aprendizaje y la implementación de esta propuesta tiene valor como objeto de estudio.

5.3. Diseño de la validación

El caso de estudio se basa en la hipótesis de que *las trayectorias de aprendizaje enseñadas mediante App Inventor son una manera efectiva de inculcar pensamiento computacional en un contexto remoto.* Según el protocolo propuesto por R. Yin para el desarrollo de un caso de estudio, las preguntas a responder se pueden clasificar en 5 grupos, según los fenómenos que se pretenden observar.

- Preguntas de Nivel 1: **Preguntas a personas específicas**. Las preguntas son respondidas por los participantes de la experiencia, o un subconjunto de ellos.
- Preguntas de Nivel 2: **Preguntas a cada caso**. Las preguntas son respondidas en base a las observaciones particulares del caso que se desarrolla en esta instancia.
- Preguntas de Nivel 3: **Preguntas sobre el patrón de los descubrimientos a lo largo de los casos observados**. Las preguntas son respondidas al observar incidencias que se repiten en varios casos que son parte del estudio.
- Preguntas de Nivel 4: **Preguntas sobre el estudio completo**. Las preguntas son respondidas por todos los elementos que conforman el estudio.
- Preguntas de Nivel 5: **Preguntas normativas sobre recomendaciones de medidas y conclusiones**. Las preguntas son respondidas por los elementos que enmarcan y delimitan el estudio

En el libro de Robert Yin [23], habla que las preguntas son para guiar la recolección de datos y que, si se tiene un caso de estudio de un único caso, se omiten las preguntas de tipo 3. Dado que nuestro caso de estudio cuenta con un solo caso (es decir, se observa un solo evento), no existen

preguntas de nivel 3. También vale la pena entender que este estudio se enmarca en una investigación mayor de enseñanza de pensamiento computacional desarrollado en la Universidad de Chile por el Departamento de Ciencias de la Computación, por lo cual hay otras experiencias con las cuales contrastar. En vista de lo anterior, las preguntas son las siguientes:

- Preguntas de Nivel 1: ¿Con qué actividades conectan más las niñas y niños que asistieron al taller? ¿Notaron los apoderados algún cambio de actitud?, ¿Notaron los docentes un cambio de actitud respecto al contenido, por parte de los niños y niñas que asistieron al taller?
- Preguntas de Nivel 2: ¿Qué tanto incorporan los contenidos las niñas y niños que asistieron al taller?, ¿Qué dinámicas se dan entre los participantes?, ¿Qué habilidades adquieren las niñas y niños que participan?, ¿La duración del taller es adecuada?
- Preguntas de Nivel 3: No hay
- Preguntas de Nivel 4: ¿En qué medida afecta que el taller sea remoto?, ¿En qué medida afecta la duración del taller la incorporación de conocimientos?, ¿En qué medida afecta que se use App Inventor por sobre Scratch?
- Preguntas de Nivel 5: ¿En qué contextos es aplicable los que funciona del taller? ¿Qué variables adicionales se pueden medir en una próxima oportunidad?

Dado que la metodología seguida para la validación es un caso de estudio, se decide recopilar información de varias fuentes para tener un mejor panorama y así entender y diagnosticar mejor. Las fuentes de información usadas para responder estas preguntas son: código generado por las niñas y niños, encuestas, observaciones ambientales, grabaciones, información sobre los alumnos y alumnas, análisis estático de código, entrevista al cuerpo docente. Para un correcto manejo y análisis, se hace uso de una carpeta Dropbox compartida entre los docentes para centralizar el material usado, subir las grabaciones de sus respectivas salas y para respaldar los archivos generados por las niñas y niños en las sesiones de trabajo. Se usa Google Jamboard como medio para llevar a cabo el análisis porque provee de una interfaz visual que simula efectivamente un tablero *kanban* con sus respectivos Post-Its, pero sin la restricción física de tamaño, haciendo mucho más fácil su uso y manejo. Los *screenshots* del estado final del tablero se encuentran en el Anexo D.

5.4. Ejecución del taller

En esta sección se reporta y discute lo observado durante la ejecución del taller, entre el 29 de mayo y el 19 de junio del año 2021. Estas observaciones vienen de analizar las grabaciones en video de las clases, las conversaciones con los tutores del curso y analizar el material producido por los participantes (código, lluvias de ideas, respuestas a encuestas). Los nombres de los participantes se mantienen anónimos, dado que se tratan de menores de edad.

A nivel de participantes, se inscribieron 29 menores en el formulario. A la primera clase llegan 20 y se tienen 16 participantes en la clase final. Si bien no se esperaba un número en particular de participantes, la reducción de asistentes de 29 a 16 llamó la atención en los tutores, pues no tenían en consideración que esto pudiese suceder. También vale la pena destacar que, a diferencia de las versiones presenciales, se abarcó una mayor diversidad de territorio: hubo alumnos de Puerto Montt y Algarrobo. Las implicancias de lo anterior es que las instancias remotas facilitan el acceso a actividades educativas a gente que suele no participar de ellas por razones geográficas.

Por el lado de los tutores, se necesitaban por lo menos dos para poder repartir 10 alumnos (de los 29 inscritos) para cada miembro del cuerpo docente (3 en total). Con la incorporación de uno más, la carga se aligeró para todos. Los tutores expresaron que la cantidad de niñas y niños con los que trabajaron fue cómoda, notándolo en la cantidad de preguntas que debían responder y el tiempo que les podían dedicar. Esto importa para el desarrollo del taller, pues se sabe por *"Choosing Face-to-face or Video-based Instruction in a Mobile App Development Course"*[4] que la presencia e interacción con tutores en tiempo real es importante para la motivación de alumnos que toman cursos remotos.

Con respecto a las encuestas de entrada y salida, fueron respondidas por 11 y 10 asistentes cada una y con 7 personas respondiendo ambas encuestas. Se les preguntó por su contexto y entorno escolar en la encuesta de entrada, en particular su relación con la computación, y su percepción de la computación. En el cuestionario de salida se les preguntó sobre su percepción del curso y nuevamente sobre su visión acerca de la programación. Las respuestas escritas responden a las preguntas planteadas, pero hubo muchas conductas inesperadas en las secciones que pedían un dibujo: la mayoría hizo *collages* de imágenes sacadas de internet y dibujos en Paint, varios entregaron la misma imagen tanto para el formulario de entrada como el de salida, un niño entregó una foto en vez de un dibujo, y justificó que no le gustaba dibujar, entre otros. . Se intuye que el número tan bajo de respuesta fue causado por varios factores: la petición de rellenarlo en tiempo fuera de la clase, la dependencia de los padres para que se enteren los niños y niñas que deben responder la encuesta y la similitud estética entre ambos, que causó que varios no respondieran la segunda porque "ya la habían respondido". Las medidas que se tomarán en una próxima instancia son dos y que atacan el problema efectivamente: dar tiempo en clases para que respondan las encuestas, las cuales serán entregadas por los tutores, y cambiar el diseño gráfico de la encuesta para diferenciarlos.

A la hora de desarrollar las clases, el entusiasmo provocado por el cuestionario en Kahoot! fue innegable: la participación se mantiene y se notaba a la hora de seguir respondiendo preguntas de la clase. Por otro lado, se tenía cierto recelo con respecto a la competitividad innecesaria que podría generar (el puntaje es asignado en base a la correctitud de la respuesta y la rapidez en responder), por lo que en la primera clase con Kahoot se hizo sin puntaje y con nombres aleatorizados (para optimizar el uso del tiempo). Dado que les extrañó mucho a los alumnos la ausencia de puntos, en la segunda clase con cuestionarios se reintegraron. No producen mayor desmotivación, por lo que se mantienen también en la sesión siguiente y se elimina la restricción de nombres aleatorios. Lo anterior cobra relevancia al identificar dos cosas que genera esta instancia: primero, es un rompe-hielo muy efectivo que aumenta la disposición a participar de la clase en todos los asistentes (lo que es valioso y fundamental en vista del trabajo de Basawapatna et al. [2] y de que las instancias remotas tienen menos participación de los asistentes). Y segundo, sirve como elemento diagnostico secundario para analizar su retención del contenido. Se observó consistencia en las respuestas correctas obtenidas, por lo que hace sentido asumir que como mínimo recuerdan de una semana a otra los contenidos trabajados.

Se les da la libertad a los participantes de mantener apagada o prender su cámara durante las sesiones, en favor de su privacidad como menores de edad. En vista de lo anterior, el espacio de enseñanza virtual fue medianamente anónimo, con sólo una porción de los asistentes dispuestos a mostrar su cara. Se le suma a esto que algunos ocupaban su nombre real y otros ocupaban alias. Se registró un caso donde el nombre mostrado era el de su mamá, por lo que existían confusiones constantes respecto a cómo llamarle. . Esto repercute en la entrega de conocimiento, pues es más fácil identificar quién preguntó si hay una conexión entre la cara y la voz, y así evitar pérdidas de tiempo al verificar en cada clase quién fue la persona que tuvo cada duda.

Durante las sesiones expositivas se registró que existían ciertos alumnos más activos que otros, mayoritariamente niños por sobre niñas. Sin embargo, en las sesiones de trabajo en grupos reducidos, esta diferencia desaparece. También se notó que, al hablar varios a la vez, prevalecían quienes hablaban más fuerte. Se les pidió que usaran la función "Levantar mano" de Zoom para hablar de manera ordenada y así escuchar a todos, pero la medida fue ignorada por los alumnos. Estas ocurrencias interrumpían el flujo de la clase, restando tiempo a la exposición o a la explicación de actividades, lo cual es crítico para una planificación con bloques rígidos para cada actividad. Se planea contar con una planificación más flexible para una próxima instancia, que pueda acoger este tipo de actitudes espontáneas.

En las sesiones expositivas se notó que divergían mucho con respecto a los conceptos, dado el enfoque cotidiano que se le da a los ejemplos. Los niños y niñas se mostraban muy dispuestos a aterrizar la conversación con experiencias propias similares a las presentadas, involucrándose así en la clase. . Este tipo de interacciones son clave para identificar que están incorporando conceptos de pensamiento computacional, pues logran extrapolar el concepto del ejemplo y aplicarlo a otros casos símiles, identificando un patrón.

Con respecto a las sesiones donde las niñas y niños deben programar, los programas provistos siempre se presentaban incompletos y esto causó un cambio en la disposición de los participantes a lo largo del taller. La actitud predominante al principio era de incertidumbre, llegando algunos a expresar que había que reemplazar el computador si algo no funcionaba. A la segunda clase del taller, les causa extrañeza que ninguno de los programas funcionen, por lo que se les revela que son diseñados así intencionalmente porque ellos son capaces de arreglarlos. Al final del taller la actitud frente a un programa mal hecho es calmada, expresando que "hay que ver lo que hace mal y arreglarlo". Si bien es algo que no se esperaba desde el principio, se identificó que este cambio de actitud es fundamental para el desarrollo de habilidades de depuración de código.

En las sesiones de trabajo, se nota que hay problemas con el manejo de la plataforma de App Inventor. Esto se evidencia en que, a la hora de buscar instrucciones para agregarlas al código, tanto tutores como alumnos deben revisar en lugares que no esperan. Tampoco logran entender del todo cada bloque por la presencia intermitente de español e inglés en la plataforma. Si bien se hizo un acercamiento durante la primera sesión, se identifican la falta de familiaridad y las interacciones inesperadas como motivos de frustración por parte de las niñas y niños. Al analizar estas incidencias, se propone que una adaptación de App Inventor a las necesidades puntuales de los niños y niñas (idioma completamente integrado al entorno de programación y guía más clara en la navegación) traería muchos beneficios en agilizar la curva de aprendizaje de la herramienta, sin considerarla poco apta para la enseñanza de pensamiento computacional por estos motivos.

Se reportaron dos situaciones inesperadas que iniciaron conversaciones al respecto entre los tutores en la retrospectiva posterior a las clases. Lo primero fue que se registró un alumno que solicita y recibe ayuda esporádica de un adulto cercano durante los ejercicios de programación. Lo segundo que se registró y causó sorpresa fue un alumno que no pudo ser parte de las actividades de programación pues el equipo que usaba contaba con restricción parental, lo que le imposibilitaba descargar los programas preparados. En ambos casos se discutió sobre qué implica eso para la adquisición de conocimiento por parte de los alumnos y cómo se puede atacar a futuro, suponiendo que situaciones así no son excepcionales. Se sugiere pedir que el computador tenga el bloqueo parental desactivado como requisito adicional al curso. También se contempla como posibilidad crear instancias (o incluso diseñar un curso entero) con el foco de que los apoderados trabajen con su niño o niña.

Se motiva a los alumnos a que usen los bloques de descanso de 15 minutos para alejarse del

computador, estirarse y comer, con los tutores dando el ejemplo al apagar la cámara y el micrófono. Sin embargo se registra que sólo la mitad hace eso. Este bloque de descanso es apreciado por los tutores, que expresan que la sesión en general fue menos agotadora gracias a que se tienen estos descansos. Lo anterior es importante porque es necesario para el correcto desarrollo del curso contar con la presencia continua y atención de los tutores para lograr un mejor proceso de aprendizaje remoto, como se expone en el trabajo de Campbell et al. [5].

Dentro de los inconvenientes experimentados a lo largo del taller se registraron como comunes la mala conexión a internet y problemas de servidor en la página de App Inventor a la hora de subir proyectos de programación o de ejecutarlos localmente en emulador. Como es un problema a nivel de servidor, se podría solucionar con una versión local de App Inventor en servidores propios. Adicionalmente, se tuvieron eventos excepcionales de corte de luz que dejó a 2 alumnos sin usar el computador durante una sesión, conectándose de manera provisoria a la clase con el celular. Esto último es importante pues hubiese sido esperable que decidieran no asistir al taller ese día, dado el contexto. Sin embargo, el asistir como oyentes es evidencia del interés y entusiasmo por el contenido entregado, lo que es fundamental para la adquisición de conocimiento según el trabajo de Basawapatna et al. [2].

Se puede reportar que en la última clase el código presentado no funcionaba, pero esta vez de manera no intencionada. La actitud de los alumnos al ver que no funcionaba fue de investigar el código y arreglarlo, en vez de declararlo un caso perdido, como si fue en la primera clase. Esta observación refuerza el cambio actitudinal hacia la programación en general y hacia los errores en particular, que en combinación abren la puerta a la adquisición de habilidades relacionadas con la depuración de código.

En las sesiones de avance del proyecto, las niñas y niños muestran entusiasmo sobre lo que van a desarrollar, aportando con múltiples ideas sobre las funcionalidades y el objetivo de la aplicación que diseñarán. El tipo de idea que presentan suele estar inspirado en algún aplicativo con el que ya estén familiarizados (más de un niño sugiere incorporar microtransacciones en el proyecto). También se reporta que el nivel de premeditación de las ideas presentadas por niñas es distinto al de los niños, siendo las ideas presentadas por ellas mucho más *ad-hoc* al contexto y las de ellos mucho mas variadas y espontáneas. Estas observaciones concuerdan con lo observado en el trabajo realizado por Grover et al. [10], relacionado al aprendizaje por desarrollo de proyectos de libre elección.

Se reporta también que, pese al entusiasmo, no avanzan en las tareas autoasignadas fuera del horario del taller. Al preguntarles a los asistentes qué causa esto, argumentan falta de tiempo para poder hacer actividades distintas a las esperadas por el colegio. Frente a esto, el diseño del taller dirige a los tutores a ofrecer programar por ellos parte de lo que hubiesen querido programar por su cuenta, en una versión modificada de *pair programming*. La recepción de esto es positivo por parte de los alumnos y se logra programar *apps* que se aproximan a los diseños que acordaron en las sesiones anteriores. Lo anterior es muestra importante de la importancia del trabajo acompañado en las instancias de aprendizaje remoto, como lo plantean Campbell et al. en su trabajo [5].

Al cierre del taller, se reporta mucha satisfacción entre los participantes. Las niñas, notoriamente, hacen énfasis en que no sabían nada de computación o no tenían cercanía con el área y que este taller les acercó a la computación. Dos participantes reportan recelo frente a quien programó las actividades (en este caso, la profesora que dictó el taller), ya que en su mayoría no funcionaban. Varios hacen énfasis en que disfrutaron mucho las clases, que no querían perdérselas y que ojalá haya una nueva instancia similar. Lo anterior muestra que se logró motivar lo suficiente para dar continuidad al curso por cuatro semanas, un desafío para un curso remoto. Vale la pena analizar lo

anterior con dos reparos: el primero es que, dado el contexto de distanciamiento social, todas sus clases de colegio tienen el mismo formato remoto, por lo que no existe un rechazo por desconocimiento de la modalidad. Además, quienes reportan al cierre del taller son quienes se mantuvieron lo suficientemente motivados para completarlo, por lo que el sesgo debe ser tomado en cuenta si se quiere aumentar la retención de alumnas y alumnos.

Por su parte, los apoderados expresaron agradecimientos al equipo de trabajo. Tres apoderados estuvieron presentes durante todas las clases del curso, acompañando a sus hijos, y expresaron sorpresa frente a la manera en que se planteaban los conceptos, relacionándolos con "cosas corrientes" para ilustrarlos. Esto es importante pues cimienta una de las ideas originales de Wing al presentar el concepto de pensamiento computacional [22]: que esta habilidad es una forma en que los humanos, no los computadores, piensan, por lo que su desarrollo es independiente de la máquina.

Un problema identificado muy tarde fue la cantidad de supuestos que están naturalizados por trabajar y desenvolverse en el área de computación en un entorno altamente académico: el manejo de inglés por parte de las niñas y niños era dispar y pone en clara desventaja a quienes no manejan el idioma, al tener nombres en inglés para la mayor parte de los conceptos asociados a la programación. Además, conceptos como indentación son naturales para quienes trabajan en el área de informática pero completamente carentes de sentido para los asistentes al taller. Finalmente, la cantidad de ejemplos necesarios para poder entender un concepto nuevo es mucho menor para alguien inserto en entornos académicos. Las situaciones anteriores describen problemas por falta de comprensión del contexto de los asistentes y pueden ser abordados en próximas iteraciones del taller.

5.5. Análisis del material

El material usado para analizar el taller son grabaciones de las sesiones de clases, tanto de la sesión principal como de las sesiones de trabajo en *breakout rooms* y grabación de la actividad de cierre del taller, donde se recoge *feedback* de los apoderados. También se usan las retrospectivas realizadas con el cuerpo docente al final de cada sesión y su respectivo registro escrito, además de las encuestas de entrada y salida completadas por los niños y niñas asistentes al taller.

En miras de facilitar el análisis de las grabaciones del taller, se revisan y se anotan observaciones en torno a tres ejes: operativo (relativo al desarrollo e implementación del taller), actitudinal (relativo a la disposición de los estudiantes con respecto a las dinámicas de la clase) y cognitivo (relativo a demostraciones de retención o comprensión de los conceptos entregados en el taller). Las observaciones son separadas entre si por color, cada uno asignado a un tutor distinto, y en paginas por día, para poder identificar fácilmente cambios o generalidades. Esta herramienta arrojó luces sobre comportamientos recurrentes que no se hubiesen notado de no ser por este registro. El tablero está disponible en el anexo D.

La grabación de la actividad de cierre se condensa de una manera similar: se agrupan los comentarios de los niños y niñas a un lado, identificados por género según el color de la nota, y los comentarios de los apoderados en otro. Esta actividad tiene la particularidad de incorporar la visión de los apoderados, lo que nos servirá para analizar el impacto del taller fuera del horario de clases.

Con respecto a las sesiones de retrospectiva al final de cada clase, estas seguían una estructura laxa de comentar los hechos relevantes o anecdóticos de la clase, compartir experiencias y hacer una

dinámica de retrospectiva (en este caso Keep-Fix-Try[1]), en pos de mejorar las próximas clases y el próximo taller. Lo anterior es importante porque presenta más puntos de vista sobre la misma situación y también aporta información de sucesos que estaban desarrollándose al mismo tiempo.

Dado que las encuestas de entrada y salida fueron desarrolladas en la plataforma de Google Forms, el manejo de los resultados ya está brindado por el entorno.

De lo anterior, se observa una gran cantidad de datos cualitativos, como intereses en cada momento, foco, intención, percepción de retención de los contenidos, entre otros. El detalle del material usado se podrá encontrar en el anexo D.

5.6. Preguntas del caso de estudio

Para responder a las preguntas del caso de estudio planteadas al inicio del capítulo, se contrastará la información del reporte y análisis del taller con distintos *papers* relacionados a la educación remota de pensamiento computacional en K-8.

5.6.1. Preguntas de nivel 1: preguntas hechas a personas especificas

- *¿Con qué actividades conectan más las niñas y niños que asistieron al taller?*

  Basados en el nivel de participación (personas distintas que responden en una actividad) observado en las grabaciones, las actividades desconectadas son las que causan una mayor reacción. Dentro de esta categoría están tanto las actividades de tipo auxiliares (el cuestionario de Kahoot!) y las de reflexión en voz alta. En segundo lugar quedan las actividades que tenían una *app* que ya funcionaba. También se notó que quienes tenían un celular o una tableta conectaban más que aquellos que usaban emulador. Esto es importante porque sustenta los avances tanto en la enseñanza remota y la enseñanza de pensamiento computacional, mediante la motivación.

- *¿Notaron los apoderados algún cambio de actitud en sus niños?*

  No se reporta un cambio de actitud generalizado durante la realización del taller. Los apoderados hablan de que notaron que sus pupilos lo disfrutaron mucho y que ellos, como apoderados, esperan que este taller sea el inicio de un interés en computación. Sin embargo, no reportan más que el entusiasmo por ir a clases, esperable para una instancia novedosa en la rutina escolar. Si bien se esperaba que hubiesen cambios actitudinales fuera de la clase, no es indicador de falta de retención de los aprendizajes.

- *¿Notaron los docentes un cambio de actitud respecto al contenido, por parte de los niños y niñas que asistieron al taller?*

  Sí, el cambio más significativo es la actitud respecto a los problemas. Al inicio eran muy pesimistas y desconfiados en sus capacidades, llegando a expresar que "esto no va a funcionar". Tampoco se sentían cómodos dando ideas para resolver un problema. Al final del taller, muchos más niños y niñas daban sus ideas. Una fuente de sesgo puede ser la reducción del tamaño del curso, que pasó de 20 a 16 alumnos, lo cual provocaría la sensación de un incremento en la

[1] `https://code-artisan.io/retrospective-method-kpt/1`

participación. Un cambio en la actitud a la hora de enfrentarse a los programas es un aporte al objetivo que apunta a desarrollar competencias de pensamiento computacional.

Una observación que podría contradecir lo expuesto es que el cuerpo docente notó falta de entusiasmo en la tercera clase. Sin embargo, no se tiene información ni inferencias sobre qué causó este cambio en ellos, por lo que no se pueden sacar conclusiones al respecto.

5.6.2. Preguntas de nivel 2: preguntas hechas a cada caso, dentro de un estudio con múltiples casos

- *¿Qué tanto incorporan los contenidos las niñas y niños que asistieron al taller?*

  Logran responder correctamente las preguntas de repaso de una clase a otra. Esto puede deberse porque incorporaron correctamente los contenidos o por la estructura de las preguntas. Se descarta que sea azar por la consistencia de las respuestas correctas.

  Logran identificarlos cuando un tutor les hace ver la conexión entre ejemplo y concepto, pero no llegan a identificarlo al programar. Dado que lo que se busca es que logren identificarlos también a la hora de programar, lo anterior nos da pistas de que las actividades deben ser modificadas en alguna medida para que logren llevar el mensaje de manera más clara.

  Logran un manejo muy básico de App Inventor (agregar elementos, identificar algunos bloques, identificar elementos de la interfaz). Se observa a la hora de conversar con ellos en torno a qué hacer en App Inventor para resolver los problemas planteados. Lo anterior es evidencia que aporta al cumplimiento del objetivo de enseñanza de pensamiento computacional.

- *¿Qué dinámicas se dan entre los participantes?*

  Se comunican y coordinan por el chat escrito de Zoom para jugar fuera del taller. También lo hacen para coordinar el avance del proyecto.

  Cuando hablan para responder una pregunta de los tutores, lo hacen todos al mismo tiempo. Incluso con la sugerencia de usar la funcionalidad de "Levantar Mano" de Zoom para que todos tengan la opción de hablar claramente, no se logra cambiar esta conducta.

  Se observa que existe confianza a la hora de preguntar cosas del taller. Se reporta que, cuando se les indica que "pueden preguntar lo que sea que no les haya quedado claro y lo repasamos desde el principio" toman la oportunidad para pedir aclaración sobre conceptos vistos en la clase o herramientas que no logran entender. A lo largo del taller, estas instancias se brindan siempre a la mitad de una actividad y al final y son aprovechadas por las niñas y niños. Esto es importante porque, tal y como se plantea en el trabajo de Campbell et al. [5], es importante contar con los tutores en las etapas de aprendizaje de una instancia remota.

  Los niños y niñas responden favorablemente a ejemplos que se relacionen con juegos o tendencias de las cuales son parte. Un caso notable fue la explicación de las "pantallas" con las que cuenta App Inventor mediante la comparación con las distintas vistas que tiene la aplicación móvil de Youtube. Con este uso de ejemplos fue muy fácil la explicación de los componentes visuales y cómo se relacionan entre si. Otro ejemplo fue que, al revisar todos los componentes disponibles en App Inventor, se relacionaron con ejemplos de usos conocidos para ellos, ya sea en juegos, páginas que frecuentan o aplicaciones que usas comúnmente.

  La interacción a la hora de dar ideas es diferenciada según género: las niñas tienden a dar ideas más aterrizadas y contextualizadas, mientras que los niños aportan de manera espontánea y sin filtro. Sorprendió encontrar estos resultados, dado que no se esperaba una diferencia en su conducta desde tan temprana edad. Este tipo de conductas se condice con lo observado por Grover [10].

- *¿Qué habilidades adquieren las niñas y niños que participan?*

  Logran ejecutar y modificar un programa en App Inventor. Esto se verifica a lo largo de las clases, cuando ya no requieren ayuda para cargar el archivo provisto y pueden navegar con mediana autonomía en la vista de bloques. Esto es un aporte a los objetivos de desarrollar pensamiento computacional, porque les permite navegar en la herramienta de desarrollo.

  Conocen el *pair programming*. Lo ejecutan de manera parcial al dictarle al tutor qué debe programar mientras ellos toman las decisiones de diseño y lógicas. Esto es un aporte a los objetivos de desarrollar pensamiento computacional, porque les permite ejercitar la abstracción necesaria para crear funciones sin tener que preocuparse de la implementación.

  Logran algo de la trayectoria de aprendizaje de depuración de programas (*debugging* en inglés). Sin intencionarlo, se pudo verificar que adquirieron conductas relacionadas a la trayectoria de aprendizaje de depuración [16], propuesta por K. Rich, C. Strickland T. A. Binkowski y D. Franklin, que también propusieron las trayectorias de aprendizaje para pensamiento computacional. La figura 5.1 es una traducción al español hecha para este trabajo de memoria y muestra la trayectoria propuesta por los autores, y encerrado a color se encuentran los logros de aprendizaje adquiridos. Se observó en distintas actitudes que lograron incorporar dos de los tres objetivos de la trayectoria para principiantes ("Los resultados pueden ser usados para decidir si hay o no erroresz Refinamiento iterativo puede ayudar a corregir errores"), y adicionalmente dos de nivel intermedio ("Pequeños errores pueden cambiar el resultado" y "Los errores pueden ser causados por información faltante en las instrucciones, en oposición a información incorrecta"). Esto es un aporte a los objetivos de desarrollar pensamiento computacional, porque les permite ampliar su arsenal de herramientas a la hora de pensar en cómo desarrollar correctamente un aplicativo.

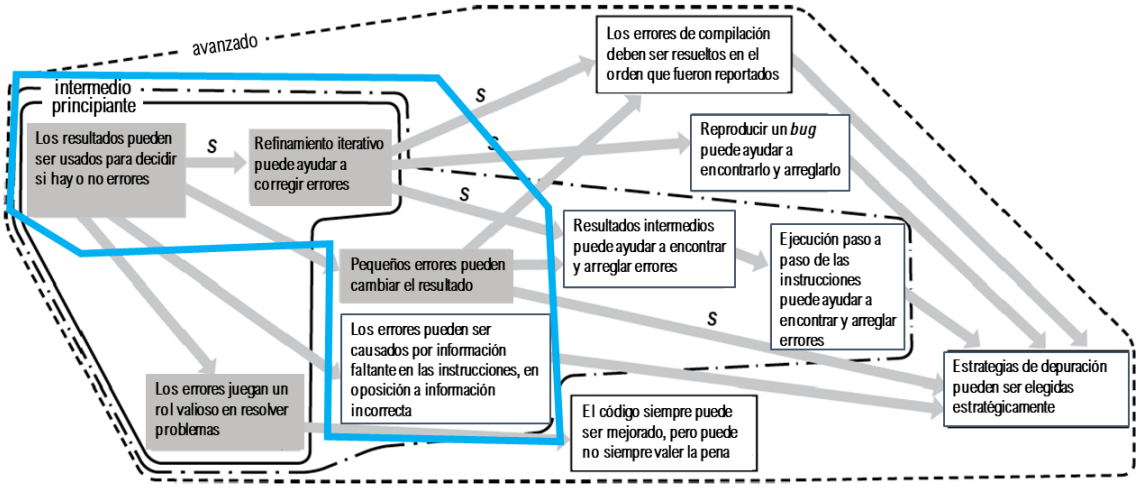

Figura 5.1: Trayectoria de aprendizaje para depuración. Traducido al español. Original tomado de *A K-8 Debugging Learning Trajectory Derived from Research Literature* [16]

- *¿La duración del taller es adecuada?*

  Según las observaciones, el tiempo no fue suficiente. El flujo de desarrollo de las actividades era la priorización de terminar de revisar una a fondo por sobre seguir fielmente un esquema de tiempo. Esto causó que algunas actividades quedaran sin desarrollarse, que algunas explicaciones fuesen meras clarificaciones o acercamientos, y no la exposición a profundidad ni reflexión con ejemplos que se tenía preparada. Faltó también suficientes tiempo para desarrollar trabajo práctico, debido a que las actividades estaban presentadas como aplicaciones incompletas y se necesitaba tiempo para hacer sentido del código en el contexto dado para resolverlo.

Según los asistentes, el tiempo no fue suficiente. A la hora de preguntarles, varios asistentes expresaron su sorpresa con que las clases durasen tan poco y/o que se impartiesen por tan pocos días. La sensación de falta de tiempo es un factor que influye tanto en el objetivo de enseñanza remota como de enseñanza de pensamiento computacional, al dedicar menos tiempo de lo que necesitan para integrar a cabalidad los contenidos.

5.6.3. Preguntas de nivel 3: preguntas sobre el patrón de los descubrimientos a lo largo de los casos observados

No corresponde en este estudio.

5.6.4. Preguntas de nivel 4: preguntas sobre el estudio completo

- *¿En qué medida afecta que el taller sea remoto?*

  Se identifican los siguientes puntos de comparación con los otros talleres que se han impartido por el Departamento:

  - Menos control sobre el desarrollo de las actividades: los inconvenientes como cortes de luz, fallas de internet o del servidor de App Inventor aumentaban la ansiedad de los niños y niñas y debían ser resueltos antes de seguir con el resto de la clase. Esto es un factor fuera del control del taller y afecta negativamente el objetivo de enseñanza de pensamiento computacional, al interrumpir el flujo incremental de la entrega del contenido.
  - Menor desgaste del cuerpo docente: dictar el curso de manera presencial requiere un esfuerzo físico distinto a la experiencia de monitorear *online* con instancias de descanso bien implementadas. Esto es importante, pues del trabajo de Campbell et al. [5] se desprende la importancia de contar con tutores en las etapas de aprendizaje de una instancia remota.
  - Posibilidad de desligar el curso de una restricción territorial: Dada la geografía de Chile, quienes viven en los extremos deben viajar mucho para tener las mismas experiencias que quienes viven en la zona centro o cerca de la Región Metropolitana, lugar donde se sitúa la Facultad de Ciencias Físicas y Matemáticas y donde se impartían los talleres anteriormente. Se ofrece la posibilidad por primera vez de participar de manera simultanea a alumnos que, según se registró, no estaban viviendo en la región. Adicional a lo mencionado, el equipo docente también disfruta de esta nueva oportunidad (al momento de dictar el taller, no todo el cuerpo docente se encontraba viviendo en la Región Metropolitana. Lo anterior es un avance en el ámbito de la enseñanza remota, al mostrar que es factible llegar a distintos lugares con este tipo de modalidad.

  Adicionalmente, llama la atención que el número de participantes que abandonaron el curso fue mayor en comparación con las instancias presenciales, pero no hay datos suficientes como para concluir.

- *¿En qué medida afecta la duración del taller la incorporación de conocimientos?*

  Se identificó la falta de ejercitación como relacionado a la poca familiaridad con App Inventor. La cantidad de tiempo destinado a programar y analizar código no es suficiente.

  Puede también que el código trabajado sea de un nivel de complejidad mayor a lo que pueden manejar programadores novatos, por lo cual la curva de aprendizaje se vuelve aún más incli-

nada. Esto repercute en el objetivo de enseñanza de pensamiento computacional, al no contar con tiempo suficiente para desarrollar completamente las actividades.

- *¿En qué medida afecta que se use App Inventor por sobre Scratch?*

  App Inventor facilita la creación de prototipos visualmente coherentes, siguiendo el esquema Modelo-Vista-Controlador (MVC). Sin embargo, al poseer tantos elementos y bloques dedicados, junto a la traducción parcial al español, se vuelve menos accesible que Scratch. Recordar que App Inventor es un entorno ideado para desarrollar *apps* de manera fácil, mientras que Scratch pretende acercar la programación a los niños y niñas, por lo que sus objetivos distan y su diseño también.

  Quienes habían ocupado ya Scratch se perdieron varias veces buscando bloques de código que no existen en App Inventor, por lo que se revela una fijación en la forma de resolver el problema. El hecho de manipular una *app* que pueden modificar ellos es un incentivo muy notorio: la mayor muestra es el interés mayor en editar una vez que se confirma que la *app* funciona en un dispositivo como un celular o una *tablet*. Quienes usan emulador muestran menos entusiasmo en desarrollar las tareas. Esto tiene un impacto en el objetivo de enseñanza de pensamiento computacional en dos ámbitos: primero, una herramienta con una curva de aprendizaje muy grande puede significar que no es la herramienta adecuada (lo cual se sabe que es falso por lo reportado en distintos trabajos [18], [7], [10]), por entorpecer la ejercitación de los conceptos aprendidos. Por otro lado, impacta positivamente al motivar el aprendizaje mediante la interactividad del resultados de la programación.

- *¿En qué medida afecta que el taller use trayectorias de aprendizaje versus otros enfoques?*

  El taller original tenía un marcado foco en la programación por sobre impartir nociones de pensamiento computacional de manera explicita. En ese sentido, el aprendizaje que se obtenía en la versión original era originado por la experimentación y el descubrimiento, no por la explicación del concepto.

  Por otro lado, el taller rediseñado tiene un foco en la explicación de conceptos y el posterior refuerzo mediante trabajo de programación o reflexión con preguntas. Adicionalmente, al ser incremental y en espiral, las niñas y niños van relacionando el material en base a lo que ya se ha visto y aprendido en las clases anteriores. Afecta directamente el objetivo de enseñanza de pensamiento computacional, al observar que el enfoque de trayectorias de aprendizaje dio buenos resultados.

5.6.5. Preguntas de nivel 5: preguntas normativas sobre recomendaciones de medidas y conclusiones

- *¿En qué contextos es aplicable los que funciona del taller?* Se logra ver que el taller tiene cualidades que le hace adecuado para el contexto que fue diseñado. Por ejemplo, las medidas de mitigación tomadas para evitar que la modalidad remota disminuyese la participación y la motivación de los participantes lo hacen idóneo para desarrollarlo a distancia. Esto es un indicador de logro del objetivo de enseñanza remota.

- *¿Qué variables adicionales se pueden medir en una próxima oportunidad?*

  Idealmente el nivel de retención de conceptos pasado cierto tiempo del taller, para medir a cabalidad si la enseñanza de pensamiento computacional es un aprendizaje que se mantiene en el tiempo. Esto afecta directamente el objetivo de enseñanza de pensamiento computacional.

Capítulo 6

Discusión e implicancias

En vista de las observaciones y resultados de los capítulos anteriores, se infiere que el curso logra motivar la curiosidad por la computación. Sin embargo se encontraron varios puntos a mejorar, lo cual es esperanzador, pues abre la oportunidad de perfeccionar lo que se tiene. Se obtuvieron muchos aprendizajes valiosos sobre la enseñanza de pensamiento computacional, acotados al contexto en el cual se desarrolló el taller. Es posible evaluar la similitud de contextos en la región geográfica cercana a Chile y aplicar lo aprendido para potenciar la enseñanza de computación, programación y pensamiento computacional con el enfoque propio de la idiosincrasia de quienes forman parte del taller.

6.1. Principales aprendizajes

De los aprendizajes que más se rescatan del taller a nivel de experiencia es que respetar el orden de las trayectorias de aprendizajes sugeridas en el trabajo de K. Rich et al. [17] fue fundamental para el correcto desarrollo del pensamiento computacional en el taller. Se observó con consistencia que los niños y niñas relacionaban lo que se estaba enseñando en el momento con conceptos presentados en clases anteriores. Además se observó que avanzar en las tres trayectorias de aprendizaje a la vez, de manera variada, es bien recibido. Queda pendiente probar una implementación de currículo con foco exclusivo en una trayectoria de aprendizaje por días, y comparar la retención de los conceptos en relación a la implementación actual.

El diseño de cada clase también fue exitoso, con cada actividad compuesta por una parte expositiva y otra interactiva, lo que brinda cierto avance con respecto a los desafíos planteados por Fagerlund et al. en su revisión de programas escolares [9] . Dado que el pensamiento computacional es una habilidad que puede ser usada más allá de la programación, vale la pena poner foco por un periodo limitado en el trabajo exclusivo de cada uno de los objetivos presentados en las trayectorias, en vez de presentar varios objetivos y luego ejercitarlos en conjunto. Se confirmó su efectividad a la hora de recapitular lo aprendido en las clases anteriores, pues tenían claridad de con qué concepto se relacionaban las preguntas y respondían correctamente.

Se observó concordancia con lo observado por Grover et al. en su trabajo [10] con respecto al tipo de aplicativo que desarrollaron, donde dos de los cuatro proyectos eran juegos y los otros dos eran

de tipo utilitario. Sin embargo, a diferencia del estudio de Grover, se les dieron restricciones sobre los elementos a usar en su *app*, por lo que se podría argumentar que no son proyectos de temática libre por completo.

Los *snippets* de código que podían ser arregladas fácilmente por los asistentes lograron a cabalidad el objetivo de motivarlos a hacer funcionar código que previamente era inútil, rompiendo la barrera de miedo a lo desconocido, lo que claramente aporta un indicio sobre cómo enseñar a niñas y niños, aplacando parte de las inquietudes planteadas por Fagerlund et al. en su trabajo [9]. Es más: lograron adquirir ciertas nociones de *debugging* y *pair programming* (cuando el tutor programaba lo que ellos querían que fuese su App), pese a no estar intencionado en el diseño del taller. Esto último se observó al comparar las nociones observadas en los niños y niñas con la propuesta de trayectoria de aprendizaje para depuración de código [16]. Se identificó que lograron 4 objetivos (presentado en la Figura 5.1), generando efectivamente un cambio actitudinal.

La actividad de Kahoot al inicio de cada clase logró mantener la energía alta desde el inicio, lo cual es un logro para un taller voluntario extra-programático y es una motivación más, como J. Davis y S. Rebelsky sugieren que se implemente en su trabajo [8]. La motivación por participar y el enfoque en el contenido por parte de los alumnos son parte fundamental del proceso de aprendizaje, como se reporta en el trabajo de Basawapatna et al. [2], y en especial es un avance en el aspecto remoto del taller, donde se había reportado anteriormente que los alumnos tenían menos conexión con las actividades en esta modalidad.

En el ámbito operativo, contar con bloques de trabajo de tiempo fijo y con descansos fijos fue valioso, sugerencia que también viene del trabajo de J. Davis y S. Rebelsky [8]. Si bien estaban orientados a aliviar la carga cognitiva y el cansancio en los asistentes, resultó ser fundamental para que los tutores lograsen desarrollar de buena manera su rol. Esto es muy importante cuando se toma en cuenta el contexto remoto en el que se desarrolla el taller, donde se reporta que la concentración y la participación son menores que en las instancias presenciales. Así, tener un cuerpo docente que puede desarrollar las actividades para motivar a los niños y niñas es imprescindible para que se logre el objetivo del taller. En vista del trabajo expuesto por Campbell et al. [5], donde se destaca la importancia de la presencia continua de los tutores en las instancias remotas para lograr los objetivos de aprendizajes planteados, se puede ver que fue una decisión que contribuyó al buen desarrollo del taller.

En vista de lo anterior, la recomendación principal es mantener el orden de las trayectorias de aprendizajes al diseñar actividades. También se recomienda enfocar el avance a nivel de objetivos de cada trayectoria de aprendizaje y no agruparlos. Las actividades de programación que se encuentran incompletas pueden ser usadas por quienes planeen un taller de programación y/o pensamiento computacional, teniendo siempre las precauciones de que sean de fácil resolución, con una dificultad incremental y de dar el espacio para que resuelvan las actividades en clase. Como se menciona más adelante, la percepción general fue que el taller tuvo muy poca duración, por lo cual parece ser provechosa la idea de alocar espacios de tiempo mayor para el desarrollo de actividades en compañía de los tutores. Esto facilitaría también la resolución de dudas con respecto a programación o a los conceptos vistos, generando un refuerzo en la materia. Finalmente la recomendación de tener segmentos de tiempo con recreos en medio y el uso de Kahoot son mejoras que cualquiera que esté planeando un taller remoto puede usar a su favor, independiente de su contenido.

6.2. Mejoras posibles

Con respecto a los elementos a mejorar, lo primero y más urgente que requiere revisión son las *apps* que acompañaban las actividades. El acercamiento de presentarlas incompletas tuvo efectos inesperados: los potenció a arreglar código y entender que es posible arreglar las cosas que no funcionan, pero también generó mucha frustración y poco acercamiento real a la programación. Se sugiere trabajar con los mismo ejemplos, pero sus versiones funcionales, idealmente con una revisión del incremento en la dificultad y completitud. Por ejemplo, para la planificación descrita en la Tabla 3.8, se pueden usar aplicaciones funcionales durante los primeros dos días, y los siguientes dos pueden desarrollarse con aplicaciones incompletas. De esta manera se genera un buen soporte para los conocimientos más básicos en el contexto del taller, según lo propuesto por Bloom [3]. Adicionalmente, se considera reformular las *apps* para que el código usado no sea difícil de navegar: comentarlo, nombrar las variables y los elementos en español y de manera adecuada a lo que representan, y simplificar estructuras e interacciones en pos de la facilidad de manejo para los niños y niñas.

Lo segundo que se notó que debe mejorarse es el tiempo en general del taller, principalmente por la poca dedicación *in-situ* a la programación de su proyecto personal, que es la instancia ideal para poder medir su retención y uso de los conceptos explorados en la clase, como se hace en el estudio de Grover et al. [10]. El tiempo tampoco fue suficiente para que los asistentes pudiesen desarrollar a cabalidad todas las actividades (responder a preguntas como "¿Qué fue lo que hicimos?", "¿Cómo lo hicimos?", "¿Cómo podríamos mejorar lo que ya hicimos?", "¿Qué cosas similares podríamos hacer ahora?", por ejemplo) y es imperativo que se debe suplir la falta de práctica en en ámbito de la programación, para lograr una mejor cohesión entre el contenido enseñado y su ejercitación. Es importante tener en cuenta que la carga académica del colegio no les permite trabajar fuera del horario de clases en las actividades propuestas por el taller, por lo que brindar un espacio en donde puedan avanzar es fundamental para que desarrollen confianza en el área de programación.

También vale la pena tener cuidado con la introducción de palabras clave como IF, ELSE, FOR y WHILE, pues, al ser palabras en inglés, no les hacen tanto sentido a las niñas y niños que asistieron al taller. Idealmente se pueden acercar como operaciones de elección, para luego ponerles un nombre, pero no se debe valer de la memorización de la palabra porque la falta de familiaridad con el lenguaje inglés juega en contra.

Finalmente, el rediseño visual del cuestionario de entrada y salida es algo útil en el ámbito operativo. Deben ser fácilmente distinguibles, para evitar que los participantes crean que ya respondieron y en consecuencia no llenen el formulario, perdiendo así información valiosa.

Aun cuando los resultados están actualmente limitados para la muestra observada, hay implicancias interesantes que permitirían sentar las bases para diseñar talleres de pensamiento computacional basado en trayectorias de aprendizajes de cursos remotos, las cuales pueden ser exploradas más a fondo en futuras versiones de este taller. Esto por la exploración del panorama actual que se realizó y la evaluación de los retos a los que se enfrenta en este escenario la enseñanza informal y a nivel K-8. Es posible refinar desde acá en adelante, y se pueden usar los conocimientos para generar una tercera versión del taller que tenga un nivel de éxito mayor, si se siguen las observaciones sobre qué mantener y qué mejorar expuestas en los puntos anteriores. Esto en si mismo es parte del proceso iterativo que es la exploración de distintos métodos de enseñanza, agregándole también el factor de distancia entre quien enseña y quien recibe el contenido.

En síntesis, se puede hacer uso de los siguientes aprendizajes para diseñar un taller de pensamiento

computacional:

- Seguir las trayectorias de aprendizajes de secuencia, condicionales y repetición propuestas por Rich et at. en su trabajo "K–8 Learning Trajectories Derived from Research Literature: Sequence, Repetition, Conditionals" [17]
- Diseñar en torno a cada uno de los objetivos de aprendizaje que entregan las trayectorias de aprendizaje, sin agruparlos
- Proveer un punto de partida para los proyectos, que sea ajeno a la temática pero relevante para el diseño, no afecta la complejidad de las ideas que se puedan presentar en los alumnos
- Hacer uso de *apps* con un nivel de complejidad creciente a medida que se desarrolla el taller
- Considerar que todo el trabajo relacionado debe ser desarrollado en tiempo de taller y planificar en torno a esa consideración

Capítulo 7

Conclusiones

En este trabajo de memoria se abordó el problema de la enseñanza de pensamiento computacional en el nivel K-8 en Chile en un contexto remoto. Anteriormente el Departamento de Ciencias de la Computación de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile dictó un Taller de Pensamiento Computacional en App Inventor para niñas y niños de 5° básico a 7° básico, pero en vista de las propuestas educacionales recientes sobre pensamiento computacional, y sumando las restricciones sanitarias vigentes, se identificó la oportunidad de estudiar una manera de entregar estos conocimientos distinta a las usadas previamente.

En este trabajo de título se rediseñó el taller de App Inventor actual en dos ámbitos: primero rediseñando el curso en torno a las trayectorias de aprendizaje propuestas por Rich, Strickland, Binkowski, Moran y Franklin [17] y luego adaptando el material a una modalidad de clases remota. Esto, para adaptarse a la situación actual de enseñanza a distancia y para hacer uso de propuestas nuevas en el área de la enseñanza de pensamiento computacional. Luego de la ejecución, fue evaluado como un caso de estudio, con preguntas orientadas a indagar la pertinencia de las trayectorias de aprendizaje y las decisiones para un taller remoto, analizando el taller en múltiples dimensiones. Se lograron aprendizajes respecto a la enseñanza de pensamiento computacional, siendo el más importante que el contenido presentado, al estar alineado a las propuestas de trayectorias de aprendizaje para pensamiento computacional, hacían sentido a los asistentes y lograron retener parte del conocimiento entregado, basado en las observaciones al curso. Además, se lograron aprendizajes sobre la enseñanza remota, como que

Para cumplir con el objetivo general de la memoria, se plantearon los siguientes objetivos específicos, los cuales fueron alcanzados por distintos medios:

1. **Identificar los parámetros relevantes para evaluar la efectividad de aprendizaje de pensamiento computacional en un contexto virtual**: Se revisó literatura relevante tanto a la educación de pensamiento computacional como de la enseñanza a distancia. Se concluyó que, si bien existe gran trabajo avanzado en ambos dominios, la intersección entre ellos es muy escasa. Adicionalmente, el área de la enseñanza de pensamiento computacional tiene propuestas recientes que son interesantes y cuya implementación tiene valor como objeto de estudio. El objetivo se cumple porque se logra construir un cuerpo de valores a medir al realizar el taller.

2. **Analizar y adaptar las plantillas de desarrollo, actividades de orientación, *snippets*,**

pautas de evaluación para el equipo docente y material de apoyo con las que cuenta el taller original para adaptarlo al taller en su nueva versión: Se adaptó en un proceso de dos etapas el material, cambiando primero el enfoque del curso por uno guiado por trayectorias de aprendizaje, y luego refinándolo en torno al requerimiento de enseñanza *online* síncrona. Se concluyó que el taller original no se alineaba a las propuestas de enseñanza en base a trayectorias de aprendizaje, por lo que era necesario el desarrollo de un plan curricular y material completamente nuevos. El objetivo se cumple al finalizar las dos iteraciones, que generan cambios adaptados a los desafíos tanto de currículo como de modalidad.

3. **Diseñar e implementar la estructura para dar soporte al taller, haciendo uso de herramientas probadas y validadas**: Se evaluaron distintas plataformas y se diseñó el aspecto operativo que daría sustancia al desarrollo del taller. Se concluyó que en ese momento, la plataforma Zoom, junto a otras herramientas tecnológicas, eran las mejores opciones para llevar a cabo el taller. Adicionalmente, se incorporó el uso de rúbricas de talleres similares para la evaluación de desempeño y resultados de los alumnos. El objetivo se logra al validar que la estructura es funcional en una etapa de piloteo.

4. **Estudiar la pertinencia de las actividades diseñadas en el marco del taller en la dimensión de actividad remota**: Mediante las preguntas enmarcadas en el caso de estudio, se observó que el taller fue diseñado de manera adecuada para el contexto remoto. Se concluyó que el taller fue exitoso porque tuvo en consideración que la presencia de los tutores es fundamental, porque para el rango etáreo con el que se trabaja la sincronicidad es clave para despejar dudas en instancias de aprendizaje y porque se consideró el cambio en la disposición a aprender en una instancia educativa a distancia. El objetivo se cumple porque las dificultades que se reportan en el uso de esta modalidad fueron mitigadas.

5. **Estudiar la pertinencia de las actividades diseñadas en el marco del taller en términos de desarrollo de habilidades de pensamiento computacional**: Mediante las preguntas enmarcadas en el caso de estudio, se observó que el taller fue diseñado de manera coherente a las trayectorias de aprendizaje, pero que puede ser refinado en este ámbito. Se concluyó que el uso de trayectorias de aprendizaje es fundamental en el éxito al integrar los conceptos los asistentes al taller. Se observó que las actividades usadas tienen espacio para mejorar. El objetivo se cumple porque los asistentes reportan retención de los conceptos varios días después de desarrolladas las clases, porque la adquisición de nuevos conceptos gatilla relaciones con los temas ya tratados y porque demuestran actitudes asociadas al pensamiento computacional.

6. **Desarrollar propuesta para la replicabilidad de este taller**: Se indican los elementos exitosos del taller, junto con los puntos a mejorar. Se concluyó que el trabajo que propone las trayectorias de aprendizaje para dar sustento al pensamiento computacional sigue una lógica efectiva, comprobada empíricamente, por lo que se recomienda como base para futuros talleres de pensamiento computacional. Se observa que las actividades deben tener un nivel de accesibilidad incremental, tanto en forma como fondo. El objetivo se cumple porque los elementos identificados pueden ser considerados para desarrollar otras instancias educativas.

Dado que los objetivos específicos son alcanzados, se puede concluir que el objetivo general de la memoria es alcanzado también.

Trabajo futuro

Con respecto a ideas que pueden mejorar el taller, se intuye que una actividad de retrospectiva sobre lo trabajado vendría a cerrar el aprendizaje de cada tema de buena manera: que deban escribir en un papel los resultados obtenidos, reflexionar sobre lo que hicieron, para realizar un barrido cognitivo. Acompañando a esto, un *handout* como en la versión original podría ser un buen incentivo al trabajo personal, al estar disponible incluso durante las horas fuera del taller.

De ser posible, montar localmente un servidor que corra App Inventor, con una traducción al español más completa. Esto serviría para mejorar la estabilidad y la experiencia en general a la hora de trabajar con los programas, bajando el nivel de ansiedad al usar la plataforma. Para ello, se requeriría un servidor dedicado y una imagen de la infraestructura. Afortunadamente, App Inventor es un proyecto de código abierto [1], por lo que es fácil cumplir con el segundo requisito. Actualmente sólo permite trabajar en emulador, es decir, sin App Inventor Companion, lo que abre otra oportunidad de mejora.

También sería adecuado generar una guía para la preparación de los tutores, donde se deje patente qué requisitos debe tener, qué conocimientos se les debe entregar *a priori* del taller y qué medidas tomar en casos generales y particulares. Esta guía reduciría la variabilidad de experiencias entre un taller y otro por contar con distinto cuerpo docente. Esto es importante a la hora de enseñar, pues genera un cuerpo mínimo de conocimiento que puede ser consultado por los tutores.

Se intuye que dar instancias para que los alumnos compartan con sus pares las maneras en que modificaron los programas en los que trabajaron puede ser enriquecedor. Se observa que los niños y niñas tienen el conocimiento suficiente sobre lo que programan como para presentárselo a otros compañeros, y el intercambio de ideas sobre qué mejorar y cómo hacerlo es beneficioso tanto como para quien presenta como para quien escucha. Este tipo de interacciones refuerza la incorporación de los hitos de las trayectorias de aprendizaje, mediante la reflexión sobre su proceso de pensamiento durante la programación.

En caso de ser positivos los resultados de las mejoras, es posible que el taller adquiera en un futuro una modalidad híbrida, incluso después de las medidas sanitarias actuales. La importancia de esta propuesta radica en que persigue el mismo objetivo del taller, pero con el desafío adicional de probar el diseño con dos grupos que reciben el conocimiento de maneras distintas. Hay desafíos en lo distinto de sus necesidades y en mantener el flujo de la clase para ambos.

[1] `https://mit-cml.github.io/appinventor-sources/`

Bibliografía

[1] Hasnan Baber. Social interaction and effectiveness of the online learning – a moderating role of maintaining social distance during the pandemic covid-19. *Asian Education and Development Studies*, ahead-of-print, 01 2021.

[2] Ashok Basawapatna, Alexander Repenning, Mark Savignano, Josiane Manera, Nora Escherle, and Lorenzo Repenning. Is drawing video game characters in an hour of code activity a waste of time? In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE 2018, page 93–98, New York, NY, USA, 2018. Association for Computing Machinery.

[3] David Reading Krathwohl Benjamin Samuel Bloom. *Taxonomy of educational objectives: The classification of educational goals*. David McKay Company, New York, 1956.

[4] Matthew Boutell. Choosing face-to-face or video-based instruction in a mobile app development course. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '17, page 75–80, New York, NY, USA, 2017. Association for Computing Machinery.

[5] Jennifer Campbell, Diane Horton, and Michelle Craig. *Factors for Success in Online CS1*, page 320–325. Association for Computing Machinery, New York, NY, USA, 2016.

[6] Douglas Clements and Julie Sarama. Learning trajectories in mathematics education. *Mathematical Thinking and Learning*, 6:81–89, 04 2004.

[7] Nathalia da Cruz Alves, Christiane Gresse von Wangenheim, and Jean Hauck. A large-scale analysis of app inventor projects. 06 2020.

[8] Janet Davis and Samuel A. Rebelsky. Developing soft and technical skills through multi-semester, remotely mentored, community-service projects. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, page 29–35, New York, NY, USA, 2019. Association for Computing Machinery.

[9] Janne Fagerlund, Päivi Häkkinen, Mikko Vesisenaho, and Jouni Viiri. Computational thinking in programming with scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, n/a(n/a).

[10] Shuchi Grover, Satabdi Basu, and Patricia Schank. What we can learn about student learning from open-ended programming projects in middle school computer science. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, page 999–1004, New York, NY, USA, 2018. Association for Computing Machinery.

[11] Francisco Gutierrez, Jocelyn Simmonds, Nancy Hitschfeld, Cecilia Casanova, Cecilia Sotomayor, and Vanessa Peña-Araya. Assessing software development skills among k-6 learners in a project-based workshop with scratch. pages 98–107, 05 2018.

[12] Francisco J. Gutierrez, Jocelyn Simmonds, Cecilia Casanova, Cecilia Sotomayor, and Nancy Hitschfeld. Coding or hacking? exploring inaccurate views on computing and computer scientists among k-6 learners in chile. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, page 993–998, New York, NY, USA, 2018. Association for Computing Machinery.

[13] Vanessa Izquierdo-Álvarez, Eva Lahuerta-Otero, and Rebeca Cordero-Gutiérrez. Kahoot, win the learning race. In *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, TEEM'18, page 737–741, New York, NY, USA, 2018. Association for Computing Machinery.

[14] Mahsa Mohaghegh and Michael Mccauley. Computational thinking: The skill set of the 21st century. *International Journal of Computer Science and Information Technologies*, 7:1524–1530, 06 2016.

[15] Tania Ponce, Cristián Bellei, and Constanza Vielma. Experiencias educativas en casa de niñas y niños durante la pandemia covid-19, 12 2020.

[16] Kathryn M. Rich, Carla Strickland, T. Andrew Binkowski, and Diana Franklin. A k-8 debugging learning trajectory derived from research literature. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, page 745–751, New York, NY, USA, 2019. Association for Computing Machinery.

[17] Kathryn M. Rich, Carla Strickland, T. Andrew Binkowski, Cheryl Moran, and Diana Franklin. K-8 learning trajectories derived from research literature: Sequence, repetition, conditionals. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*, ICER '17, page 182–190, New York, NY, USA, 2017. Association for Computing Machinery.

[18] Krishnendu Roy. App inventor for android: Report from a summer camp. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, SIGCSE '12, page 283–288, New York, NY, USA, 2012. Association for Computing Machinery.

[19] Neil Smith, Mike Richards, and Daniel G. Cabrero. Summer of code: Assisting distance-learning students with open-ended programming tasks. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE 2018, page 224–229, New York, NY, USA, 2018. Association for Computing Machinery.

[20] Patrick Wang. Pseutopy: Towards a non-english natural programming language. In *Proceedings of the 17th ACM Conference on International Computing Education Research*, ICER 2021, page 429–430, New York, NY, USA, 2021. Association for Computing Machinery.

[21] Robert Whyte, Shaaron Ainsworth, and Jane Medwell. Designing for integrated k-5 computing and literacy through story-making activities. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*, ICER '19, page 167–175, New York, NY, USA, 2019. Association for Computing Machinery.

[22] Jeannette M. Wing. Computational thinking. *Communications of the ACM*, 49(3), 2006.

[23] Robert K. Yin. *Case Study Research: Design and Methods*. SAGE Publications, 2003.

Anexos

A. Anexo A

Material usado en el taller actual (incluye una carpeta con el material del taller original): `https://bit.ly/3AX97YI`

B. Anexo B

Lista de requisitos para los participantes:

Requisitos de insumos:

- Computador con Chrome, Firefox o Safari como navegador
- Acceso a celular o tablet Android del año 2011 o superior
- (Android OS 2.1 o más)
  o
- Acceso a iPhone del año 2015 o superior (iOS 9.5 o más)
- Acceso a cuenta de google
- Micrófono

Requisitos Participantes:

- Previa participación en taller de Scratch
  o
- Estar cursando 5to, 6to o 7mo básico

C. Anexo C

Correo enviado con guía de instalación adjunta :

Estimados!
Les envío la guía de instalación para App Inventor. En ella se cubre lo necesario para que el taller sea experimentado de la mejor manera por los asistentes. En total, la instalación demora máximo media hora. Frente a cualquier duda, no duden en escribirme!
También les adjunto el link para la clase de mañana y las subsiguientes: http://bit.ly/taller-appinventor Partiremos mañana a las 10:00 y terminaremos a las 12:30. La estructura de la clase es de dos bloques de 45 minutos y uno de 30, con descansos de 15 minutos entre cada uno. Les recomiendo a los tutores que faciliten el descanso en los bloques de 15 minutos con colación y conversación, siempre y cuando sea posible.
Finalmente, recuerdo que es imprescindible para mi trabajo de memoria que los consentimientos informados sean respondidos por todos los asistentes.
Sin más que agregar, les agradezco muchisimo su cooperación Saludos y nos vemos mañana!
Karina Parga Valenzuela

D. Anexo D

Formularios de inscripción, consentimiento informado, formularios de entrada y salida; tablero de análisis de clases y cierre de taller: `https://bit.ly/39PVWNg`

E. Anexo E

Primer cuestionario de Kahoot!: `https://bit.ly/3kY368D`

F. Anexo F

Segundo cuestionario de Kahoot!: `https://bit.ly/3ik8UqX`

G. Anexo G

Tercer cuestionario de Kahoot!: `https://bit.ly/39UNWe0`