



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE COMPUTACIÓN

Mate Marote: plataforma educativa de juegos para el entrenamiento de competencias cognitivas

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Matías López y Rosenfeld

Director: Dr. Diego Fernández Slezak
Codirectora: Lic. Andrea Paula Goldin
Buenos Aires, 2012

MATE MAROTE: PLATAFORMA EDUCATIVA DE JUEGOS PARA EL ENTRENAMIENTO DE COMPETENCIAS COGNITIVAS

La Argentina está viviendo una oportunidad única con una gran cantidad de alumnos de escuelas primarias y secundarias recibiendo sus propias *netbooks* provistas por programas institucionales, tal como el Programa Joaquín V. González en La Rioja. La existencia de esta plataforma común sumado a nuevos descubrimientos de la neurociencia aplicados a educación abre un nuevo panorama en el diseño de prácticas educativas permitiendo intervenciones a nivel masivo. En esta tesis se presenta Mate Marote, un *framework* flexible para intervenciones de escala provincial. Esta plataforma permite incorporar juegos y actividades educativas, y está diseñada para un mantenimiento y actualización automáticos. Los juegos y actividad puede registrar las estadísticas de uso y enviar dicha información para un posterior análisis. La primer prueba piloto fue puesta en producción en La Rioja, donde se instalaron manualmente más de 100 computadoras. Esta prueba piloto incluyó tres juegos para el entrenamiento de capacidades cognitivas: planeamiento, memoria de trabajo y control inhibitorio. Los resultados preliminares muestran que el comportamiento observado en esta intervención, a pesar de que no ser supervisada, son consistentes con los resultados de las intervenciones anteriores sujetas a supervisión por partes de los instructores, sugiriendo la posibilidad de escalar este proyecto. Esta experiencia abre una nueva forma de estudiar, e intervenir, en el desarrollo cognitivo con la posibilidad de una profunda reformulación en el campo de la enseñanza y el aprendizaje.

Palabras claves: Juegos de computadora, Intervención Educativa, One Laptop Per Child (OLPC), La Rioja, Neurociencia educacional, Mate Marote, Cognición

MATE MAROTE: EDUCATIVE FRAMEWORK GAMES FOR COGNITIVE TRAINING

Argentina is living a unique opportunity where a great amount of primary and secondary school students have their own laptop, given by institutional programs such as Joaquín V. González (La Rioja). The fact that all learning and teaching tasks are represented in the same digital environment is a formidable tool for educational applications, as well as for research and experimentation. From the perspective of information technology, the challenge is how to interface with this vast corpus of cognitive development data to convert conceptual questions relevant to educational practice into quantifiable, analytic queries. We designed a flexible framework consisting of educational games and activities, designed for automatic maintenance, keeping up-to-date information of this application and registering anonymous usage behavior for further analysis. The first pilot intervention was deployed in La Rioja, with more than 100 computers installed. This deployment consisted of three games to train cognitive capabilities: mainly planning, working memory and inhibitory control. Interestingly, even though this intervention is unsupervised, preliminary statistic analysis shows that children gaming behavior is consistent with previous supervised-interventions results. This effort suggests a novel window for studying (and intervene in) human cognition development, with a great amount of new possibilities, pushing a profound reformulation of teaching and learning.

Keywords: Computer Games, Educational Intervention, One Laptop Per Child (OLPC), La Rioja, Educational Neuroscience, Mate Marote, Cognition.

AGRADECIMIENTOS

Hace ya unos 10 años que arranqué la facultad. Fue un largo camino en el cual pasaron muchas cosas que llevan a que un día como hoy esté escribiendo los agradecimientos de mi tesis. Siempre pensé que iba a ser algo fácil, sencillo, lo primero que me iba a salir. Pero ahora me doy cuenta lo mucho que he vivido y la gran cantidad de gente a la que me gustaría agradecer el haberme acompañado en este largo camino.

En primer lugar quiero agradecer a la familia toda, la política y la de la vida. En especial a Álvaro y Diána por todo el amor y las grandes ayudas que me dieron durante estos 29 años que llevo de vida. A Pepín y a Andre por todos los grandes momentos que vivimos juntos y los que vendrán. A Glo, por estos 4 años bancandome. A Pancha y Sancho. A mis tíos, abuelos y primos, en especial Laloni y Cristina, la Yaya y el Tata, Netty y Natalio. A la familia que descubrí que tenía escondida en Núñez: Churruca, Celina, Mariela, Pablo y Feli. ¡Gracias por tantas noches compartidas! ¡Por hacerme sentir un poco menos lejos de mi familia semana a semana durante muchos años! Sobre todo, gracias Feli por esas grandes comilonas en las que tuve el honor de poder estar.

Un poco más allá está otra parte de la familia, la de los amigos. En particular quiero agradecer a Nacho y Ali, simplemente por ser lo que son. A Dani, que aunque está viviendo lejos creo que es una de las personas que más confió en mí y me empujó para avanzar en la carrera. A Tommy, con quien hice la mayoría de las materia con el plan de “a una por cuatrimestre eventualmente nos recibimos”, es cierto Tommy, metele que vos también ya estás ahí. A Juan, por tantas noches de charlas y momentos memorables. A Facu, por todos los grandes momentos compartidos. A Román, por los más de 30 recitales que debemos haber compartido en todos estos años y los incontables litros de mate *sentados al cordón de la vereda* (y en otros lugares también). A Píter, por ser Píter. Y a todos aquellos con los cuales he compartido la carrera o grandes momentos en la facu, y seguramente de más de uno me olvidaré: Pachi, Fran Laborda, Fran Giménez, Dani Foguelman, Hernán, Germán, Gutes, Herman, David, Tebi, Cristian, Pablo, Mariano Moscato, Dani Villani, Mica, Nano, Ale, Fede, Eleze, Jota, Gáby, Euge, Vero, Leti, Guido, Celes, Sol, José y algunos a los que ya les perdí el rastro como Nico Pajoni y Tiger.

A todos los integrantes del LNI, todas grandes personas con las que he compartido divertidos almuerzos, seminarios en los que no entendía nada (y algunos en los que empecé a entender), clases de ajedrez, experimentos y hasta asados en el Tigre. Gracias Mariano, Ceci, Alejo, Luz, Juli, Diego, Agustín, Luciano, Ariel, Julie, Martín, Juan, Andrés, Joaquín, Pablo.

A mis compañeros docentes, con los que aprendí muchísimo y que estoy orgulloso de haber laburado a su lado: JPG, Charly, Schapa, Diego Bendersky, Nico Rosner, Fran Eijó, Esteban Mocos, Ramiro Rela.

A los que nos unió el amor a la natación. Diego Sintas, Caro, Noralí, Ailín, Carlitos y Mabel, Kary, Jorgito, Ricky, Dante, Lucila, Eli, Esteban Carli.

A esa otra gente, que está por fuera de todas las categorías anteriores pero que no por eso es menos importante, Nair, Paula, Pablo, Guille Mattei, a toda la gente de la DOV, a los Divus, a la gente de DualWeb y de NSNC, a los chicos del Túnel, a Ponce y sus charlas, a Chulio, a Maranga.

A Graciela, por los incontables termos de mate y las milanesas recargadas de fritas

que fueron mi fuente de energía todos estos años.

A la bondiola. A la Abuela Lidia.

A Ana, Pancho y Nubis, los personajes de Mate Marote, que me acompañan pegados en mi termo y me tiran buena onda.

En especial un gran agradecimiento a Diego y Andre, que la verdad me tuvieron infinita paciencia y me bocha ayudaron para que esta tesis llegue hasta donde llegó.

A Álvaro y Diána.

*Al Tata, quien me hubiese encantado que
pudiera estar para verme en este momento...*

Índice general

1..	Introducción	1
2..	Mate Marote	5
2.1.	Las actividades de Mate Marote	5
2.1.1.	Control inhibitorio (Avioncito)	5
2.1.2.	Planificación (Casitas)	6
2.1.3.	Memoria de trabajo (Memomarote)	7
2.2.	Implementación	8
2.3.	Experiencia en las escuelas	9
3..	Desarrollo	11
3.1.	Diseño	12
3.2.	Implementación	14
3.2.1.	Implementación del cliente	15
3.2.2.	Implementación del servidor central	22
3.2.3.	Bases de datos	23
4..	Resultados	28
4.1.	Prueba piloto: La Rioja	29
4.1.1.	Monitoreo de actividad	30
4.1.2.	Análisis de estadísticas de uso	32
5..	Conclusiones	36
6..	Trabajo a futuro	38
	Apéndice	40
	Apéndice	40
A..	Financiamiento	40
	Bibliografía	41

1. INTRODUCCIÓN

En la última década, los países tecnológicos punteros en el mundo [1] comenzaron a insistir con la necesidad de aprovechar los recursos aportados por la Web y las herramientas computacionales para así lograr un rendimiento educativo más acorde con el entorno real en el que se desarrolla la vida cotidiana de muchos niños y jóvenes.

En la educación tradicional, el docente es el encargado de enseñar y monitorear continuamente el desarrollo cognitivo introspectivo de los alumnos. Con la incorporación de la tecnología, hoy es posible seguir el proceso de aprendizaje de los niños mediante la aplicación de algoritmos de análisis sobre las trazas que van dejando en su interacción con la computadora.

Como resultado del crecimiento de esta temática, surge una nueva disciplina llamada neurociencia educacional (*educational neuroscience*), en la que se utilizan métodos de la neurociencia cognitiva y comportamental para su aplicación en intervenciones educativas [2]. Estas estrategias consisten, entre otras cosas, en investigar el desarrollo de representaciones mentales y su evolución a lo largo del proceso educativo e intentar utilizarlas para mejorar algunos aspectos de la docencia, utilizando conocimientos preexistentes de la neurociencia.

La identificación de las rutinas mediante las cuales resolvemos problemas resulta muy relevante en la práctica educativa. De la misma manera que conocer la arquitectura de un procesador permite optimizar el código para ejecutar programas, conocer la arquitectura detrás del cómputo humano podría permitir la optimización de la actividad educativa. Por ejemplo, nadie sabe bien cómo hacemos para reconocer la emoción de una cara, para leer un texto o para hacer aritmética. Esta dificultad radica en que los algoritmos que subyacen estas operaciones no son explícitos, es decir que no somos conscientes de ellos, por lo que no podemos reportarlos. Sin embargo, mediante métodos indirectos es posible dilucidar los procesos involucrados en la solución de problemas.

Por ejemplo, un proyecto interdisciplinario de investigadores de la ciencia cognitiva y de la computación ha llevado a cabo un programa de ingeniería inversa para detectar las operaciones involucradas en el cálculo aritmético. Una gran serie de estudios desarrollados durante varias décadas ha mostrado la existencia de un sistema de cálculo aproximado, de carácter universal [3]. Este sistema está presente a los pocos días de vida [4], y es ubicuo en todas las culturas (aun en aquellas que no tienen ningún tipo de escolaridad matemática) [5] y está presente aun cuando esté opacado por un sistema simbólico. Como demostración sencilla, cualquier persona tarda más en responder si $52 > 31$ que si $52 > 17$ [6]. Si bien desde un punto de vista formal ambas operaciones son equivalentes, en un sistema analógico la segunda resulta más sencilla, y esta huella es observable –de hecho, es inevitable– aún en personas con gran formación matemática [7]. De la misma manera, nociones de geometría (aún geometrías no euclídeas sobre una esfera) son universales y están presentes mucho antes de ser formalizadas y simbolizadas en la educación formal [8].

La capitalización de estas ideas a través de juegos e intervenciones informáticas basadas en nuestro conocimiento de la arquitectura del procesador humano (el cerebro) es parte de un esfuerzo creciente en estos días [2] y está en gran sincronía con los objetivos de esta tesis.

Hoy en día vivimos en una era digital, en donde una fracción significativa de los es-

tudiantes y docentes poseen sus propias computadoras, constituyendo un fenómeno social que está produciendo un cambio revolucionario en la educación. En este sentido, muchas iniciativas están siendo implementadas entregando computadoras de bajo costo a alumnos y docentes de escuelas primarias y secundarias. En particular en la Argentina esta iniciativa sigue dos estrategias diferenciadas en cada uno de los niveles. Para el nivel secundario, la iniciativa está enmarcada en un plan nacional: Conectar Igualdad [9]. Este plan contempla la entrega de una computadora a cada alumno de escuela secundaria de gestión pública, de escuela de educación especial y de institutos de formación docente de todo el país y es llevado a cabo por el estado nacional. Actualmente se encuentran entregadas aproximadamente 2.500.000 de máquinas. Para ver un resumen del estado de entrega en tiempo real, puede accederse al sitio oficial, [10].

Por el contrario, en el caso del nivel primario cada provincia tiene libertad para elegir su propia iniciativa. Por ejemplo, la Ciudad Autónoma de Buenos Aires por medio del Plan Sarmiento ha elegido las computadoras Intel Classmate [11] para el abastecimiento de equipos para los alumnos. En la provincia de La Rioja, en el marco del Programa Joaquín V. González, han adoptado el programa de One Laptop Per Child (OLPC, *Una Computadora Por Niño*) dirigido por N. Negroponte [12] para todos los alumnos de la escuela primaria y sus docentes, alrededor de unas 60.000 máquinas. OLPC es un proyecto cuyo objetivo es el desarrollo de una tecnología de bajo costo para que cada niño tenga su propia máquina. Para poder tener costos bajos, estas computadoras, llamadas XO, se venden únicamente en grandes cantidades (del orden de los 10.000 equipos), razón por la cual sólo son compradas por programas institucionales. Entre los fundamentos de por qué es importante que cada niño tenga su propia computadora está la concepción de que una *netbook* es para un niño una ventana y una herramienta. Una ventana al mundo y una herramienta con la que pensar, abriéndoles la posibilidad a aprender a través de la interacción independiente y exploración con las computadoras. Este programa se encuentra ampliamente difundido con más de 2.500.000 de computadoras XO repartidas en todo el mundo. En latinoamérica, varios países incorporaron las XOs en sus propios programas: Perú, Colombia, Paraguay y Uruguay. El programa Uruguayo, Ceibal, es el más grande, cubriendo todo el país (primaria y secundaria). Se entregaron más de medio millón de XOs, o *ceibalitas* como las llaman en Uruguay, además de instalar puntos de acceso a Internet en todas las escuelas del territorio uruguayo.

El Plan Ceibal busca promover la inclusión digital con el fin de disminuir la brecha digital, tanto respecto a otros países como entre los ciudadanos de Uruguay, de manera de posibilitar un mayor y mejor acceso a la educación y a la cultura. La sola inclusión de la tecnología en las escuelas no asegura el cumplimiento de la meta si no se la acompaña de una propuesta educativa acorde a los nuevos requerimientos, tanto para los alumnos como para los maestros. Por lo cual el Plan Ceibal se basa en un completo sistema que busca garantizar el uso de los recursos tecnológicos, la formación docente, la elaboración de contenidos adecuados, además de la participación familiar y social. Los principios estratégicos que encierra este proyecto son: equidad, igualdad de oportunidades para todos los niños y todos los jóvenes, democratización del conocimiento y del aprendizaje, no sólo en lo que respecta a la educación que se les da en la escuela, sino también aprender ellos mismos a utilizar una tecnología moderna. Este mismo enfoque se está siguiendo en La Rioja.

Por otro lado, la incorporación masiva de las nuevas tecnologías han abierto la puerta a nuevas formas de aprender y estudiar. Un claro ejemplo de esto es la Khan Academy [13],

que propone que las clases no sean dadas por una persona sino grabadas en un video que el alumno puede ver tantas veces como quiera, frenando, retrocediendo, volviendo a revisar, y que el docente esté para ayudar a la comprensión de algún tema específico que le genere dificultades. A su vez, propone como alternativa a las evaluaciones tradicionales un seguimiento diario mediante ejercicios realizados en una computadora. En este esquema cada alumno avanza a su velocidad y dedica el tiempo que sea necesario para cada tema.

Ceibal inauguró en su plataforma Web en junio de 2011 una sección que contiene videos de la Khan Academy traducidos al español, para que puedan ser vistos por todos los alumnos uruguayos desde sus *ceibalitas* [14].

La idea de usar la computadora en el sistema educativo no es novedad y esta presente desde hace más de 20 años [15]. Los sistemas de tutorio inteligente comenzaron en la década del 80 y consisten en programas de computadora con el objetivo de proveer instrucción individualizada a los usuarios sobre temas específicos, de forma automática, sin la intervención humana.

La educación de los alumnos con seguimiento por parte de tutores humanos ha mostrado ser muy efectiva para el proceso de aprendizaje [16,17]. Evidentemente, es imposible pensar en una educación con tutores para cada uno de los alumnos, por lo que la utilización de tutores inteligentes automatizados (*intelligent tutoring system* - ITS) resulta muy atractiva. Existen muchos ejemplos de ITS usados tanto en ambientes educativos como profesionales. Uno de los más paradigmáticos es la inclusión de un ITS de álgebra en la escuela secundaria [18]. En las últimas décadas, la inteligencia artificial ha avanzado hasta el punto donde es posible construir ITS que son tan eficientes como los tutores humanos [19].

Con los programas que entregan *netbooks* se logra tener una plataforma digital común para la enseñanza y el aprendizaje. Esta tecnología permite el estudio de la interacción de los niños con sus máquinas tanto en clase como en el hogar, accediendo de esta forma a la observación de los procesos del cómputo humano asociados. Uniendo estas ideas de disponibilidad digital y educación basada en evidencia proveniente de los productos del pensamiento, se abre una posibilidad única de práctica educativa que potencialmente podría abarcar la población de una provincia o país incorporando métodos novedosos en la enseñanza.

En esta tesis se presenta una herramienta de intervención educativa de escala masiva. El objetivo principal consiste en el desarrollo de una plataforma general para la realización de intervenciones que permitan, a través de la tecnología actual, el monitoreo del proceso de enseñanza y aprendizaje a gran escala, i.e. a lo largo y a lo ancho del país.

Esto trae aparejadas grandes ventajas y, a la vez, grandes desafíos. Por un lado, contar con estadísticas de uso de las actividades a niveles masivos, que permitan la identificación de marcadores estereotipados del cómputo humano. Pero a su vez trae aparejado un gran desafío: el desarrollo de métodos automáticos de análisis de contenido provenientes de la minería de datos capaces de extraer esta información a partir de estas estadísticas de uso.

En comparación con los esfuerzos previos mencionados: por un lado los esfuerzos en secundarias de Khan Academy y por otro los esfuerzos en escuelas seleccionadas realizados con ITS, se propuso llevar estas iniciativas a edades más tempranas de escolaridad (primeros años de primaria en lugar de educación secundaria o terciaria) lo que conlleva problemas teóricos y prácticos novedosos. Una gran diferencia respecto de esta iniciativa es que la herramienta que se desarrolló fue pensada en escalas provinciales y nacionales (cientos de miles de niños), en vez de un grupo experimental reducido como se ha hecho anteriormente. Esta masividad permitirá llegar a datos más robustos que convaliden, o no,

los resultados obtenidos durante la primera etapa del Proyecto Mate Marote (detallado en el capítulo 2), salvando la diferencia de que en esta nueva versión, los niños van a jugar de manera no supervisada y sin restricciones: jugarán en la casa, en la escuela o en alguna plaza; mucho tiempo o poco tiempo; solos, con compañeros, con familiares o como ellos quieran.

El trabajo presentado se basa en resultados preliminares de la versión anterior del Proyecto Mate Marote. Esta primera versión consistió en hacer jugar a alumnos de 6 y 8 años a juegos en una computadora bajo la supervisión de un instructor y durante el horario escolar, en escuelas seleccionadas. Estos juegos apuntan a estimular capacidades cognitivas específicas, por ejemplo la memoria de trabajo. Tomando como base estos juegos, se diseñó la herramienta y realizó una intervención piloto en La Rioja, aprovechando el programa Joaquín V. González de entrega de computadoras OLPC a todos los alumnos de escuela primaria.

2. MATE MAROTE

Mate Marote es un proyecto de intervención educativa para la estimulación cognitiva a través del uso de juegos en computadora. Surge inicialmente como una prueba piloto en escuelas primarias de la Ciudad de Buenos Aires y alrededores, en las cuales se utilizó una combinación de diferentes juegos distribuidos en Módulos de Entrenamiento Cognitivo (MEC). Estos MEC se enfocan en entrenar distintas capacidades cognitivas. En particular en este caso son:

- Control inhibitorio
- Planificación
- Memoria de trabajo

2.1. Las actividades de Mate Marote

Las actividades implementadas se llaman: Avioncito, Casitas y Memomarote. Cada una de estas implementa un MEC: control inhibitorio, planeamiento y memoria de trabajo respectivamente.

Los destinatarios de las actividades son niños de los primeros años por lo cual las actividades están planteadas sobre un entorno agradable y colorido en el cual hay tres personajes (Ana, Pancho y el gato Nubis) que participan en cada una de las actividades. En Avioncito, están Pancho y Nubis que juegan con avioncitos de papel y sobre estos avioncitos lanzados por Pancho se desarrolla el juego. En Casitas los 3 personajes participan, hay que ayudarlos a que cada uno llegue a su casa. Por último en Memomarote, los personajes van a la playa y participan apareciendo en las fichas del juego cual si de fotos de vacaciones se tratara.

A continuación se detallan los tres juegos orientados a entrenar procesos cognitivos de control inhibitorio, memoria de trabajo y planificación que componen Mate Marote hasta el momento.

2.1.1. Control inhibitorio (Avioncito)

Este juego está basado en tareas de flexibilidad cognitiva y control inhibitorio estilo stroop [20] y busca entrenar la capacidad de inhibir una respuesta dominante para conseguir una respuesta subdominante.

Los participantes tienen dos opciones de dirección (derecha o izquierda) y deben decidir, lo más rápidamente posible, cuál es la correcta. En algún lugar visible de la pantalla aparece un avión de papel y el niño debe indicar una dirección mediante el teclado. Si la dirección elegida es la derecha, debe presionar la tecla **L**; si es la izquierda, la tecla **A**. Los aviones pueden ser amarillos o rojos y estar mirando a la derecha o a la izquierda. Si son amarillos, la dirección que hay que elegir es la misma a la que apunta el avión; si el avión es rojo, en cambio, hay que elegir la dirección opuesta. Por ejemplo: si el avión es amarillo y mira hacia la derecha, hay que presionar la tecla **L**; pero, si el avión es rojo y mira hacia la derecha, hay que presionar la tecla **A**.

En el centro de la pantalla, además, hay un reloj que indica el tiempo máximo para resolver el ensayo. Si se supera ese tiempo, el ensayo se considera perdido y comienza uno nuevo. En la porción superior de la pantalla hay una barra que avanza y que indica el nivel del juego. En la porción inferior de la pantalla, una barra que avanza indica el tiempo máximo de juego total. El juego se termina cuando a) se cometen tres errores dentro de un mismo nivel o b) cuando la barra inferior llega al extremo derecho de la pantalla. En cualquiera de los dos casos, el niño puede comenzar un nuevo juego.

El primer nivel es de familiarización con la tarea y presenta sólo avioncitos amarillos. A partir del segundo nivel aparecen los aviones rojos.

Este juego toma la imagen común a todos los juegos: Pancho y Nubis juegan con avioncitos de papel y la dirección que se debe indicar depende del color y la dirección de dichos avioncitos.

2.1.2. Planificación (Casitas)

El juego de planificación (entendida como armar un plan de acción para alcanzar un objetivo determinado) se basa en la prueba Dog, Cat, Mouse diseñada por Klahr [21].

En este juego hay tres personajes (un niño, Pancho; una niña, Ana; y un gato, Nubis) y tres lugares (casas), que pertenecen a cada uno de ellos. Los personajes y los lugares están dispuestos sobre un tablero cuadrado, que tiene cuatro caminos paralelos a cada lado y un camino diagonal entre sus partes superior izquierda e inferior derecha (ver Figura 2.1). En

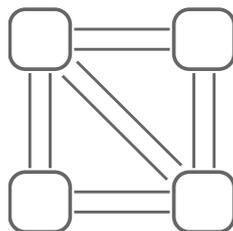


Fig. 2.1: Tablero del juego Casitas. Las casas de los personajes se ubicaran en 3 de los 4 vértices y luego los personajes sobre los mismos. Los vertices están unidos por caminos, hay 4 sobre las aristas del tablero y uno en la diagonal.

las esquinas del cuadrado se encuentran los cuatro lugares válidos. Los personajes pueden moverse a lo largo de los caminos en forma bidireccional, de a uno por vez y completando el recorrido de punto a punto.

El objetivo de un ensayo es llevar a cada personaje a su casa, desde una configuración inicial de lugares –diferente a la de los personajes– hasta una configuración final de personajes igual a la de los lugares. El espacio de problemas de todas las posibles configuraciones iniciales y de todos los posibles movimientos para llegar a la configuración final forma un grafo conocido que permite analizar las trayectorias durante el juego (ver Figura 2.2).

Cada movimiento se hace mediante *drag and drop*. Si un personaje es soltado fuera de una posición final válida o en la misma posición de la cual partió (si el niño se *arrepintió* antes de finalizar la movida), a los efectos del análisis la acción no se contabiliza como una movida. Las tres reglas del juego son: los personajes deben moverse de a uno por vez y siempre hasta un espacio vacío, los personajes sólo pueden moverse a través de los puentes y no pueden compartir casa.

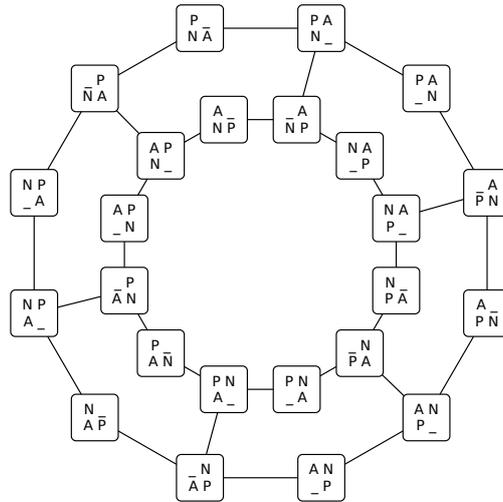


Fig. 2.2: Espacio del problema del juego Casitas. Cada nodo del grafo representa una configuración del tablero, dónde la A es Ana, la P es Pancho, la N es Nubis y _ es un lugar vacío. Cada eje une dos configuraciones indicando que se puede pasar de una a la otra moviendo un personaje. Si consideramos que los personajes tienen un orden relativo a la posición que ocupan y este orden se da recorriendo los vértices en sentido horario, entonces el círculo externo y el interno representan las dos particiones de tableros que son alcanzables realizando movimientos de personajes sin utilizar la diagonal, mientras que el pasaje de un círculo al otro implica que un personaje se desplazó por la diagonal.

El juego presenta un amplio rango de niveles de dificultad. Cada ensayo se define en función de la longitud de la trayectoria –cantidad de movimientos necesarios para alcanzar el objetivo–, que puede variar entre uno y siete movimientos. La menor distancia utilizada fue de dos movimientos y, luego de tres ensayos correctos consecutivos, el número de movimientos aumentaba en uno. Además, cada nueva sesión comienza en el mismo nivel en el que cada niño había terminado la sesión anterior.

El juego está dividido en dos fases. Una primera etapa de exploración libre, en la cual los niños conocen el número mínimo de movimientos que deben hacer para ganar pero, si se exceden, no pierden: todos los ensayos se ganan. Esta fase comprende tres ensayos para cada cantidad de movimientos (dos a siete inclusive). La segunda etapa, consecutiva a la anterior, es de movimientos restringidos: los ensayos sólo se consideran correctos si se juegan de forma óptima, si se resuelven en la mínima cantidad posible de movimientos. Esta fase comienza con al menos tres ensayos consecutivos de dos movimientos.

2.1.3. Memoria de trabajo (Memomarote)

La memoria de trabajo refiere a la capacidad de almacenar y manipular información por períodos muy cortos de tiempo [22]. El juego de memoria de trabajo se basa en un paradigma que requiere memoria de reconocimiento para patrones visuales pero no para ubicaciones espaciales.

Cada ensayo implica varias jugadas y consiste en un tablero y en un cierto número de fichas distribuidas aleatoriamente en él. Cada ficha es una imagen definida por una serie

de atributos. Las fichas siguen con la temática común, cada una contiene atributos de la temática playa, es decir: baldes, sombrillas, personajes con indumentaria de playa, etc.

Los atributos pueden estar presentes o ausentes (por ejemplo, una ficha puede, o no, tener baldes) y, cuando presentes, pueden asumir un amplio rango de valores. Los atributos incluyen color de fondo (puede ser un color liso o la imagen de una playa), forma de la ficha, sombrilla (en diferentes colores o ausente), cantidad de estrellas, cantidad de baldes, personaje (pueden ser tres en diferentes posiciones, o ninguno).

Cada ensayo, constituido por varias jugadas, tiene por objetivo seleccionar (clicar) todas las fichas que aparecen en el tablero. Luego de que el niño selecciona una ficha, las fichas desaparecen y reaparecen, después de 2 segundos, en una ubicación diferente (siguen siendo las mismas fichas pero están mezcladas). En la siguiente jugada, el niño tiene que seleccionar una ficha no elegida previamente. La dinámica continúa hasta que el niño selecciona, sin repetir, todas las fichas o hasta que comete un error. Cada vez que el niño selecciona una ficha anteriormente elegida, se computa un error y pierde el ensayo. Cada vez que el niño hace una selección completa correcta (gana el ensayo), el número de atributos posibles aumenta para la siguiente jugada (se “complejizan” las fichas) y, cada cinco ensayos consecutivos correctos, el número de fichas también aumenta. Si el niño pierde tres ensayos consecutivos, el número de fichas disminuye. Además, cada nueva sesión comienza con el número de ítems con el que ese niño terminó la sesión anterior.

En cada ensayo, cualquier atributo puede ser definido como binario asumiendo sólo uno de dos valores. Así, esta variable binaria puede ser potencialmente utilizada por los niños para agrupar y categorizar los ítems. Las estrategias de búsqueda de los niños son registradas por el juego, lo que significa que, en un futuro, el juego podrá identificar automáticamente errores para cada nivel de dificultad, así como estrategias utilizadas por los niños, y actuar en consecuencia.

2.2. Implementación

Uno de los condicionamientos para la implementación del programa fue que estuvieran diseñados de forma de poder utilizarlos en las computadoras existentes en las escuelas. Al no poder establecer una distribución homogénea de máquinas en las distintas escuelas, se implementaron los MEC utilizando tecnología Web.

En los casos en que las escuelas tuvieran buena conexión a Internet, los juegos podrían jugarse desde el servidor central instalado en la FCEN-UBA. En las escuelas sin conexión, o conexión inestable se procedería a instalar un servidor local que proveyera los juegos los días correspondientes. La tecnología elegida para el servidor Web fue Django [23] encargado de proveer los juegos implementados en JavaScript. Se puede acceder a los juegos de Mate Marote en la página oficial del proyecto [24].

Teniendo en cuenta que esta implementación sería utilizada por alumnos de los primeros años de escuelas primarias, se diseñó una interfase propia que involucra personajes de dibujos animados y un guión común, culturalmente pertinentes. Las actividades implementadas en esta prueba piloto tienen un diseño similar que incluye un algoritmo para la adaptación continua del nivel de dificultad basado en el desempeño del niño. En particular, en este caso fueron planteadas como juegos en los que los niños tienen que ir pasando una serie de niveles y a medida que aumenta su rendimiento en los juegos, aumenta la dificultad a la que se enfrentan, pero cuidando siempre de que la dificultad del juego no le sea imposible al niño. Esta esquema de complejidad adaptativa en el que la dificultad

cambiará en tiempo real en base al desempeño de cada niño es semejante al implementado en el MEC diseñado por Wilson y colegas [25].

En cada módulo del programa se registran las respuestas de los alumnos así como también los tiempos de reacción para luego ser analizados cuantitativamente. Para decidir si efectivamente los MEC entrenan ciertas capacidades cognitivas, se efectúan evaluaciones a cada niño antes y después de su participación en los sucesivos encuentros en los que se utiliza el software educativo. Estas evaluaciones también consisten en la presentación de problemas a resolver en el formato de juegos de computadora. El grupo control es un grupo de niños que, con la misma frecuencia y condiciones, juega a tres juegos diferentes, todos sin demanda cognitiva específica pero muy atractivos y con un compromiso motor semejante a los MEC utilizados (juego 1: un niño salta entre lianas, se usa la barra espaciadora; juego 2: una carrera de autos en la cual hay que tomar decisiones entre dos caminos, no restringida por tiempo, se usa el *mouse*; juego 3: una bola debe saltar entre rocas para no caer por un precipicio, se usan las flechas del teclado).

En todo momento, instructores capacitados estaban con los niños para explicarles las actividades y para asistirlos en caso de ser necesario.

2.3. Experiencia en las escuelas

Durante los años 2008 a 2011 se visitaron escuelas de la Ciudad Autónoma de Buenos Aires y el Gran Buenos Aires para la realización de la intervención educativa.

En el año 2008 participaron 12 niños de 1^{er} grado y 24 de 3^{er} grado, en el 2010 111 niños de 1^{er} grado y en 2011 fueron 90 también de 1^{er} grado. Debido a la epidemia de gripe AH1N1, durante el año 2009 no se pudo realizar actividad alguna, ya que el ciclo lectivo fue modificado y no pudieron realizarse las actividades contempladas en el proyecto Mate Marote. De haber existido la forma de que los niños jugaran en sus casas, durante el año 2009 se podría haber realizado una intervención. Pero la modalidad adoptada para la misma, presencial en las escuelas, no lo permitió.

Cada intervención consistió en una serie de sesiones en las cuales los niños, supervisados por docentes previamente capacitados para tal efecto, salían del aula para sentarse en una computadora en la cual jugaban entre 15 y 20 minutos a alguno de los juegos.

Se puede observar un resumen de la actividad en la tabla 2.1 en el que se detallan para cada año los grados con los que se trabajó y la cantidad de niños.

Año	Grados	Cantidad de niños
2008	1 ^{ero}	12
	3 ^{ero}	24
2009	Sin actividad en escuelas por la gripe AH1N1	
2010	1 ^{ero}	111
2011	1 ^{ero}	90

Tab. 2.1: Actividad en las escuelas del programa Mate Marote

La intervención realizada fue muy costosa, tanto en lo que refiere a recursos como a tiempo requerido para cada una de estas sesiones.

Debido a estos factores es que se decidió ampliar la intervención a escala masiva (utilizando las computadoras entregadas a los alumnos por OLPC) a la simple provisión de

juegos educativos sin necesidad de supervisión, lo que dio como resultado el trabajo presentado en esta tesis.

3. DESARROLLO

Como se mencionó en la introducción, hoy en día existen diversos programas de entrega de computadoras para alumnos de educación primaria y media, tanto a nivel nacional como provincial. En particular, esta tesis estuvo concentrada en el programa Joaquín V. González de la provincia de La Rioja. Dicho programa es parte de un proyecto integral en el que, además de la entrega de *netbooks* para alumnos, se facilita también el acceso a Internet instalando puntos de acceso en escuelas y espacios públicos de uso completamente gratuito.

Este programa adoptó la tecnología del programa OLPC como plataforma a utilizar en todas las escuelas primarias de La Rioja. Las computadoras de OLPC, también llamadas XO, utilizan como sistema operativo una distribución de Linux llamada Fedora [26]. Además, como las XOs están pensadas para niños, se diseñó una interfaz gráfica sencilla llamada *Sugar* [27], que se basa en principios cognitivos y constructivistas, y atractiva que reemplaza el escritorio estándar.

Sugar es el primer proyecto que creó una interfaz de usuario que se basa en conocimientos y en socialización: los alumnos deben participar en la exploración y colaboración. Se basa en tres principios muy simples:

- todo el mundo es un maestro y un alumno
- los seres humanos por su naturaleza son seres sociales
- los seres humanos por su naturaleza son expresivos

Estos son los pilares de una experiencia de usuario que acelera el aprendizaje. *Sugar* también considera dos aforismos:

- se aprende haciendo. Entonces, si querés aprender más, hacé más
- el amor es mejor maestro que el deber, uno se engancha con las cosas que le son auténticas, las que quiere.

El contacto con otras personas siempre está presente en la interfaz de *Sugar*: la colaboración es lo primero. Los usuarios participan en un diálogo mutuo, apoyándose unos a otros, pudiendo aportar y compartir críticas e ideas. Cualquier par de máquinas puede conectarse y trabajar juntas con alguna aplicación que permita trabajo cooperativo, o simplemente compartir archivos e incluso compartir aplicaciones. Esto último es tan fácil como 2 clics con el *mouse*.

Sugar está basado en el lenguaje interpretado Python. Se cuenta con el código fuente de las aplicaciones incluidas (lo cual permite modificarlas para introducir mejoras), y además es simple crear e incorporar nuevas aplicaciones apuntando a la incorporación directa de ideas, que pueden desarrollarse en la misma computadora.

Una de las cosas a resaltar es la estética que utiliza para presentar distintas pantallas. La gran mayoría está basada en íconos y colores, por ejemplo, uno puede buscar la computadora de un amigo que se encuentre cerca y reconocerla por el color que dicho amigo le dio a su usuario.

Teniendo en cuenta estos valores de contorno, se propone el desarrollo de un *framework* de juegos para intervenciones educativas de gran escala que utilice esta plataforma. El objetivo es por partida doble: por un lado para la estimulación de competencias cognitivas; por el otro, se registrarán estadísticas de uso de los juegos para la detección de nuevos factores que permitan identificar nuevas capacidades a entrenar o correlaciones entre el resultado de distintos juegos y que, a su vez, permitan mejorar los juegos de modo de conseguir que la estimulación sea cada vez más específica y eficiente.

3.1. Diseño

Se propone la creación de una herramienta flexible y escalable para intervenciones educativas. Dichas intervenciones serán realizadas a través de juegos y actividades que se incorporen a la herramienta propuesta. Estas actividades estarán definidas (o adaptadas) según la población para la que estén diseñadas, tanto en edad como región. Por ejemplo, en nuestro caso nos concentramos en niños de entre 5 y 7 años para el entrenamiento de memoria de trabajo, planeamiento y atención con tres juegos creados para tal fin.

Los usuarios de juegos de computadoras van perdiendo el interés en ellos si presentan errores, si se tornan repetitivos o monótonos [28]. Por lo tanto, se propone una herramienta que permita la actualización automática al tener conexión a Internet, agregando nuevas funcionalidades, arreglando *bugs* o agregando juegos a lo largo del tiempo sin que requiera, para su instalación o actualización, la intervención de alguien capacitado para hacerlo. Adicionalmente, si la herramienta pretende servir para intervenciones masivas, ésta debe permitir la incorporación de actividades sin la necesidad de tener que instalarlas en cada máquina.

Parte del proyecto incluye el registro de estadísticas de uso para poder analizar el comportamiento de los usuarios. A partir de este análisis se pretende descubrir marcadores que permitan mejorar los juegos actuales y el diseño de nuevas actividades para incentivar competencias cognitivas basadas en el comportamiento observado en los niños.

Teniendo en cuenta las limitaciones de conectividad, estas estadísticas de uso serán guardadas localmente en la máquina hasta que ésta se conecte a Internet. Al conectarse, dichos datos deberán ser enviados a un servidor de almacenamiento y luego borrados de la computadora local. Esta característica tiene un alto grado de importancia en cuanto a la seguridad de la información: todos los registros serán tomados en forma anónima, con el único objetivo de detectar tendencias poblacionales y de ninguna manera evaluadoras de un individuo. Este trabajo ha sido evaluado y aprobado por el Comité de Ética de la Dirección de Investigación del Centro de Educación Médica e Investigación Clínica “Norberto Quirno” (CEMIC) [29], Unidad Asociada del CONICET, bajo el Protocolo n° 486, del 4 de agosto de 2008 y reprobado el 31 de agosto de 2012.

Esta base de datos anónima de productos del pensamiento, obtenidos a través de la herramienta desarrollada y aquí presentada, será puesta a disposición para investigaciones de temas afines, por ejemplo educación. Un aspecto muy importante, debido a las características del proyecto, es la confianza por parte de la comunidad acerca del manejo de datos de estadísticas de uso. Además de contar con la aprobación del comité de ética, se proveerá el código de forma abierta y gratuita para su utilización y auditoría.

Para el diseño de esta herramienta, se identificaron los actores involucrados y las características particulares de cada caso. A partir de este análisis se extrajeron los requerimientos asociados al uso del *framework* por parte de estos actores. Luego, se procedió al

diseño de la infraestructura del proyecto, teniendo en cuenta los requerimientos relevados. Por último, se pasó a la implementación de las distintas piezas de software involucradas.

A continuación se detalla una descripción de los usuarios del *framework*:

- **Usuarios de los juegos.** Como se explicó anteriormente, la herramienta está diseñada para intervenciones educativas, principalmente enfocadas en intervenciones en la escuela primaria, por lo cual los usuarios serán niños de 5 a 13 años. En particular, en este trabajo, la intervención es en niños con edad entre 5 y 7 años, participantes del programa Joaquín V. González de La Rioja (Argentina). Aquí destacamos algunas características presentes en estos actores:
 - Poseen dificultades para leer o incluso no saben aún, principalmente aquellos que se encuentran en primer grado, que recién están empezando su educación primaria, por lo cual debería tener poco texto y el que tenga debe estar en imprenta mayúscula, ya que facilita la lectura.
 - Se distraen fácilmente, lo cual es natural en los niños, esto lleva a que los juegos tengan que ser atractivos y divertidos si uno quiere que los niños jueguen a ellos. Se buscó lograrlo utilizando siempre los mismos personajes, Ana, Pancho y Nubis, y con dibujos con formas redondeadas y de colores vivos. Esta idea de personajes proviene de varios años de trabajo de diseño por profesionales, con *feedback* de niños. Además, se incorporó una barra de nivel, para que los niños puedan ver su avance y esto los empuje a mejorar su rendimiento en los juegos.
 - No se puede suponer que tengan acceso continuo a Internet. Aunque el programa Joaquín V. González contempla que todas las computadoras tengan acceso a Internet en todas las escuelas y plazas, todavía no están habilitados todos los puntos de acceso. Por ejemplo, algunas escuelas del interior de la provincia aún no están conectadas. Ésto lleva a que todo el funcionamiento de este *framework* no pueda depender de la conexión, por lo que deberá estar completamente instalado localmente en las *netbooks*, descartando la posibilidad de una implementación Web de este proyecto.
 - Si bien prácticamente toda la población posee su propia computadora, es de esperar que los niños presten su *netbook* a compañeros, amigos o hermanos. En el caso de que éste utilice el *framework*, estaría modificando el estado de los juegos del dueño original, por ejemplo alterando puntajes máximos, pantalla actual de un juego, avatar preferido, etc. A partir de esta premisa, se decidió soportar varios usuarios locales que almacenen los datos propios de configuración y avance de los juegos.
 - Se consideró relevante que un usuario del *framework* pueda cambiar de máquina (debido a rotura, actualización, reinstalación por mantenimiento, etc.) sin perder sus datos de avance y configuración. Lo cual sugiere la existencia de algún sistema de usuarios que pueda almacenar perfiles y, por la naturaleza del contexto, que el sistema sea Web parece una buena alternativa.
 - Si bien en una primera etapa, la prueba piloto, podemos identificar un único perfil de usuario (niños de 5 a 7 años), el diseño deberá contemplar múltiples tipos de usuarios como pueden ser *docente* y *alumno*, para que puedan tener distintas actividades, vistas, etc. Otros perfiles podrán ser definidos, por ejemplo, en función de su ubicación geográfica o edad escolar, lo que permitiría que niños

de escuela primaria de La Rioja tengan aplicaciones distintas (o variaciones de las mismas) a las de niños de la misma edad que viven en Uruguay y éstos, a su vez, distintas de las actividades y juegos para los alumnos de secundaria de Uruguay.

■ **Desarrolladores de actividades.**

- Deben lograr que todas las actividades sigan una misma línea estética y un mismo estilo de programación, necesidad que se cubre presentando una serie de recursos que permiten estandarizar sus productos:
 - Registro de eventos de forma estandarizada para un posterior análisis.
 - Almacenamiento de sesiones de juego y variables de configuración para poder restaurar el estado de un juego, incluso en caso de que se cierre el *framework*.
 - Herramientas de construcción de pantalla y afines a la interfaz.
- Deben poder identificar a cada máquina que se comunica con el servidor. Esto debe permitir poder tener control sobre el flujo de datos a través de la red. Es deseable, además, poder generar estadísticas sobre el servidor que permitan analizar cantidad de conexiones, tráfico en determinados intervalos de tiempo, ubicación geográfica de las computadoras que acceden, etc.
- Deben asegurar que el *framework* tenga la capacidad de conectarse, enviar los datos registrados y actualizarse automáticamente. Este es uno de los puntos críticos del desarrollo ya que una vez instalado en las computadoras no se tendrá más acceso físico a ellas como para reinstalar o actualizar el *framework* y sus actividades. Además debe ser un proceso automático, es decir, que no requiera ninguna acción por parte del usuario para realizarse.
- Todo lo desarrollado debe ser multiplataforma. Si bien esta primera prueba piloto fue principalmente pensada para OLPC, sabiendo de la existencia de otros programas educativos de similares características (una máquina por alumno, Internet en escuelas y plazas, etc.), la implementación debe ser lo más portable posible.
- Es deseable que las intervenciones puedan provenir de distintos grupos de educadores o investigadores. Por lo que se deberá poder poner a disposición de aquellos que deseen participar del proyecto con algún juego o aplicación, el acceso a la infraestructura propuesta.

3.2. Implementación

Teniendo en cuenta los aspectos comentados en la sección anterior, se implementaron dos piezas de software que trabajan en conjunto, por un lado el servidor central y por otro lado, el cliente. Por cliente se debe entender que es la implementación del *framework* que se instalará en las máquinas de los niños que en el caso de esta prueba piloto, será en las XO's. Mientras que por servidor central se entiende a aquella implementación que provee los servicios que permiten mantener actualizados a los clientes, además de recibir sus estadísticas de uso.

3.2.1. Implementación del cliente

El cliente fue implementado con las siguientes características:

- Para el desarrollo del *framework* se contó con dos equipos XO-1.5, que es el modelo actual entregado por OLPC. Como breve descripción de este equipo se puede decir que cuenta con 1GB de memoria RAM, 4GB de almacenamiento (extensible a 32GB con una tarjeta SD) y un CPU de 1000 MHz x86 VIA C7.

Estos equipos no tienen especial énfasis en su potencia sino en su usabilidad, razón por la cual estas máquinas tienen carcasas muy resistentes, pantalla que permite utilizarse al sol sin inconvenientes. Sin embargo, para mantener el precio, el *hardware* terminó resultando no muy potente. Esto llevó a que sea necesario optimizar partes del código reduciendo operaciones y teniendo precalculados algunos valores.

- Con el objeto de que sea portable se utilizó el lenguaje *open source*, multiplataforma y gratuito: Python. Fue elegido por tener intérpretes en los sistemas operativos más usados hoy en día y por contar además con librerías para realizar interfaces gráficas orientadas a juegos como Pygame y Pygtk.
- Se desarrollaron herramientas para la estandarización de las actividades:
 - Registro de datos de uso: se provee una función para almacenar registros durante el uso de los juegos: `log`. Esta función almacena de manera consecutiva estos datos en un archivo específico para cada usuario y para cada actividad. Cada registro realizado incluye automáticamente la fecha y hora en la que fue realizado. De esta manera se asegura a los desarrolladores de actividades que utilizando esta función, los datos almacenados serán enviados al servidor central de manera automática.
 - Almacenamiento de datos de juegos: a su vez también se proveen funciones que permiten guardar datos y recuperarlos en la misma sesión de juego o en otra posterior. Nuevamente esto es transparente a los programadores, que deberían únicamente usar las dos funciones provistas a tal fin: `load_data` y `save_data` que de manera automática leen, o graban, en un archivo correspondiente al jugador que se encuentra activo para el juego en curso.
 - Recursos gráficos: se ofrecen ya implementados recursos para juegos como una barra de nivel, personajes varios, botones, puntajes, pantallas predefinidas, pantallas de instrucciones, gestor de eventos de teclado y de mouse, etc.
- Como se deben guardar datos sobre sesiones, juegos instalados, versiones y demás, se incorporó una base de datos SQLite [30]. Este fue el motor utilizado debido a que no requiere ningún tipo de instalación complementaria ya que trabaja con archivos directamente, además tiene librerías para ser utilizadas desde Python.
- Una de las características que el *framework* debía incluir es la capacidad para tener varios usuarios dentro de una misma máquina. Por lo tanto se incorporó un sistema de usuarios en el cual cada usuario puede poner su nombre y una serie de datos más, como edad, sexo, cuál es su mano hábil, etc. En la pantalla principal del *framework* hay un botón que permite cambiar de usuario y uno que permite crear un usuario nuevo. Los usuarios se crean con un pequeño formulario.

- Además de la existencia de usuarios, se desea también poder *migrar* un usuario de una máquina a otra, si bien esto quedó fuera de la implementación de esta primera prueba piloto. Está contemplada y proyectada la figura *webuser*, que es un usuario del *framework* que además tiene un correo electrónico y una contraseña asignada, y para el cual se guardan sus datos en el servidor central (y, por lo tanto, requiere conexión a Internet) para permitir recuperar el perfil desde una máquina nueva tal como estaba en la original.
- Para poder controlar qué computadora realiza pedidos al servidor central y qué información le será devuelta, se decidió implementar un protocolo de sesiones muy simple: antes de que una máquina realice algún pedido al servidor, debe autenticarse enviando su número de serie. Este número de serie es algo con lo que cuentan todas las XOs pero, llegado el caso de que se use en otra computadora, de no conseguir el número, enviará la dirección MAC (*media access control*) de la placa de red. Como respuesta al envío se le da un número de sesión. Dicho número es el que utilizará como identificador a lo largo de la comunicación con el servidor central. Se puede ver un ejemplo del uso en las Figuras 3.1 y 3.2.
- Para que el *framework*, de manera automática, se mantenga actualizado y envíe las estadísticas de uso que registró, se implementaron dos rutinas que se ejecutan cada vez que se inicia el *framework*, si este está con conexión a Internet, llamadas *upload* y *update* que se ejecutan de manera consecutiva en ese orden. Para todas las transferencias se manda:
 - *upload*: cada vez que se abre el *framework* con conexión a Internet, se ejecuta la rutina de *upload* de las estadísticas de uso. Inicialmente se inicia una nueva sesión con el servidor. Acto seguido, se comprimen en un archivo formato *.zip* los datos de los perfiles existentes y se envían al servidor. Posteriormente, para cada uno de los juegos instalados se chequea si tiene estadísticas de uso, las que también son comprimidas, en caso de existir, a un archivo *.zip* y enviados al servidor. Cuando se recibe confirmación de que fueron recibidos correctamente, se borran de la computadora. Luego se procede a ejecutar la rutina *upload*. Este proceso puede observar en detalle en el diagrama de la figura 3.1.
 - *update*: cada vez que se abre el *framework* con conexión a Internet, luego de ejecutar la rutina de *upload* el *framework* iniciará la rutina de *update*. Ésta consiste en iniciar una sesión con el servidor, para luego realizar dos etapas de actualización: la del *framework* y la de los juegos.

La primera etapa es la del *framework*, en ella se envía la versión de *framework* que se encuentra instalada y como respuesta se recibe un mensaje que informa que existe una versión nueva o que está en la versión correcta. Si hay una versión nueva, procede a descargarla y actualizarse para luego pasar a la segunda etapa. Si no hay actualización, pasa directamente a la segunda etapa.

En la segunda etapa se verifican las versiones instaladas de los juegos. Primero se pide la lista de juegos que el *framework* debería tener instalados. Para cada juego de dicha lista, se envía el número de versión instalada, o si es un juego no instalado, se informa esto y recibe el aviso de que hay una versión por instalar o que no hay actualización para este juego. En caso de que suceda esto último, simplemente se pasa al siguiente juego. En cambio, si sí tiene versión

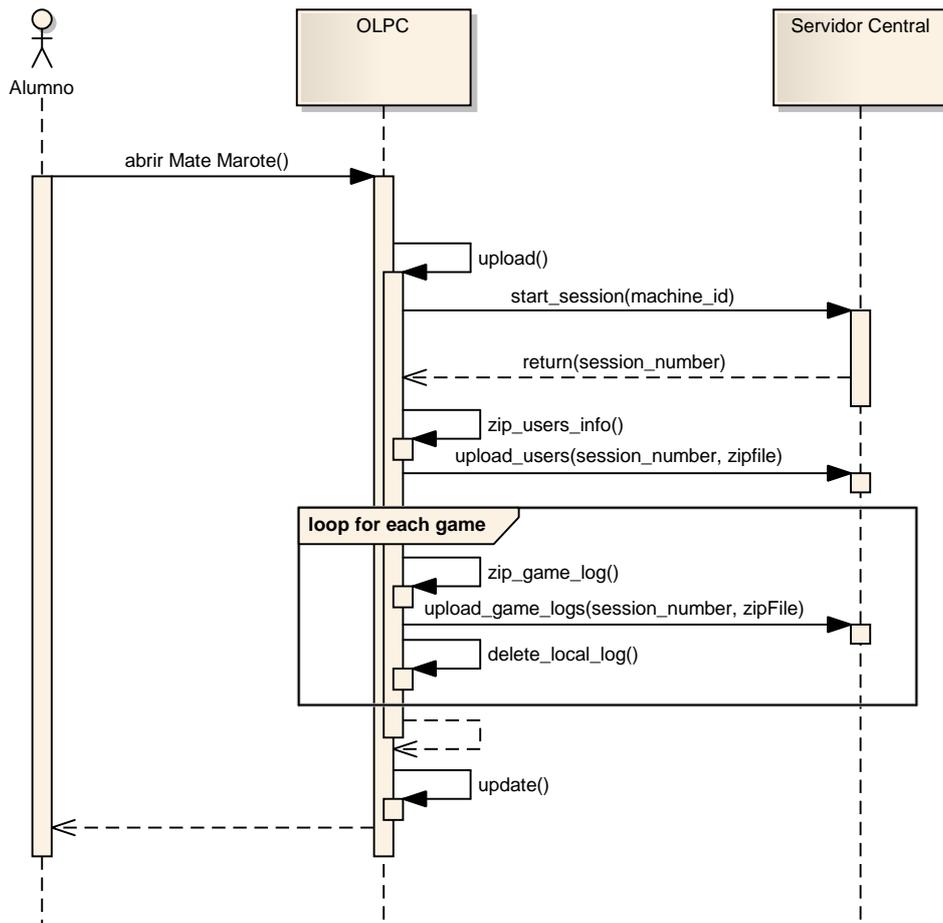


Fig. 3.1: Diagrama de secuencia de un *upload*. Al iniciar el *framework* con conexión Internet, se inicia una sesión y se procede a enviar los datos de los perfiles comprimidos en un archivo formato *.zip*. Luego se envían, comprimidas de la misma manera, las estadísticas de uso de los juegos instalados en la *netbook*.

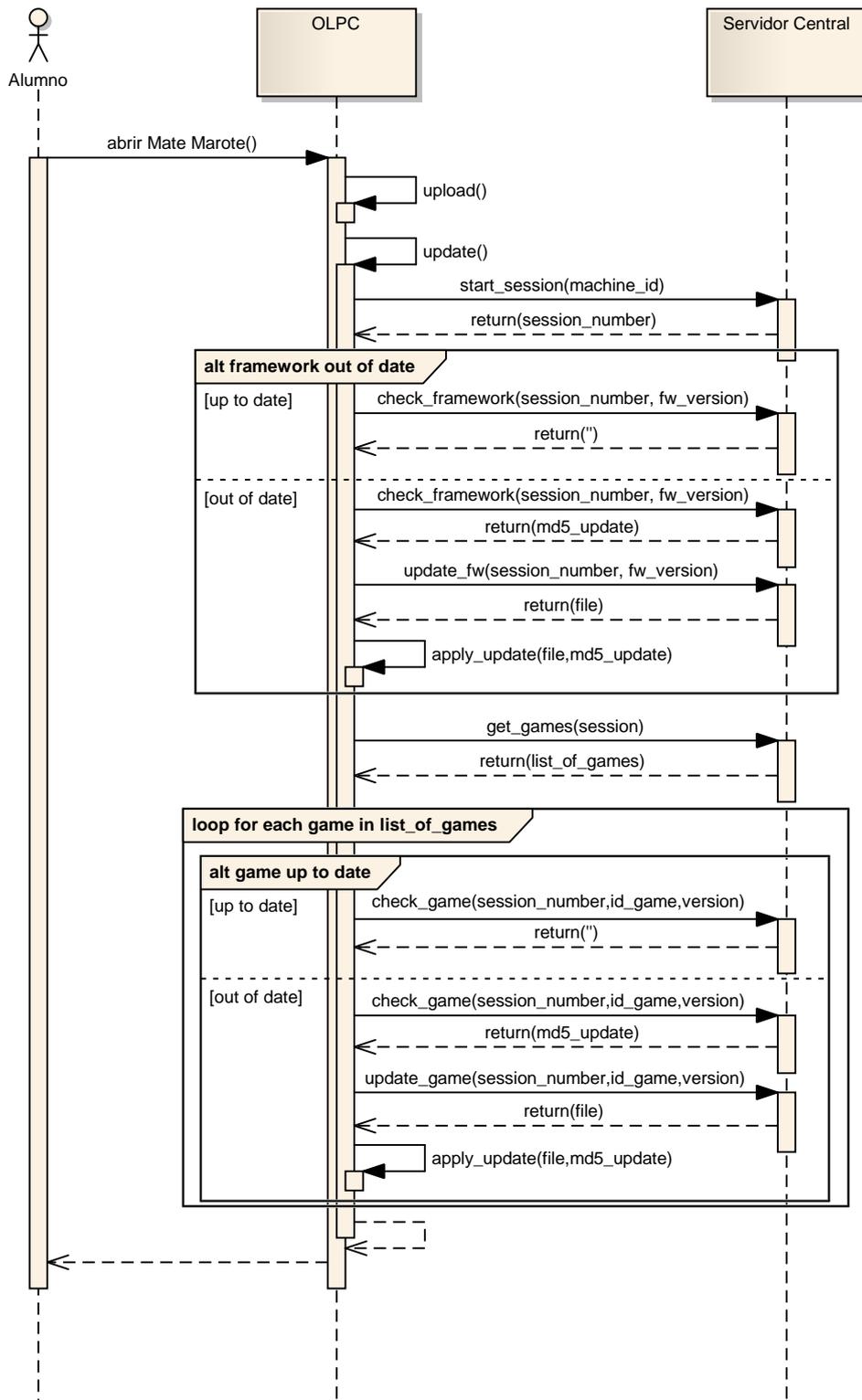
por instalar, se pide la actualización al servidor y actualiza el juego para luego seguir con el siguiente juego de la lista.

Una vez que todos los juegos se encuentran en la versión correcta, el *framework* está listo para usar.

El proceso de *update* se puede observar con detalle en la figura 3.2.

Dado que no se podía saber de antemano con qué tipo de conexiones se iba a estar trabajando, se incorporó MD5 como chequeo de integridad de los archivos: antes de enviar un archivo, se envía su *md5* y, al recibirlo, se verifica la integridad del mismo, si no se encuentra corrupto, entonces se utiliza, si no, se descarta y se trata como si no se hubiese recibido nada por falta de conexión.

Por otro lado, para poder asegurar que todas las máquinas se pudieran actualizar en cualquier lugar con Internet, se implementaron todas las transferencias vía Web, dado que la política de filtrado podría llegar a ser muy estricta en las escuelas. Con lo cual se optó por utilizar urls en el puerto 80, ya que es el mismo que tienen que

Fig. 3.2: Diagrama de secuencia de un *update*

tener habilitado para poder navegar en páginas web, actividad que consideramos que estará permitida en la mayoría, si es que no en todos, los puntos de acceso.

- Para el registro del comportamiento de los niños jugando, se utilizó la función `log` presentada anteriormente. Para cada juego se estableció el formato a utilizar para su posterior análisis. En todos los casos se registrará el inicio del juego y el estado del juego en general entre cada ensayo.

Para Avioncito en particular, se guarda por ensayo: el tipo de avión (color y sentido) y la respuesta posible: el niño no responde (`NO ANSW`), responde correctamente (`CORRECT`) o responde erróneamente (`WRONG`).

Se puede observar en detalle un extracto del archivo en el cual se guarda el registro de estadísticas de uso en el Listado 3.1. A continuación se detalla lo expresado en

```

1 [2012-05-21 16.37.20] INIT INICIANDO JUEGO
2 [2012-05-21 16.40.19] Status Correct: 0 - Wrong: 0 - Continuos correct: 0 - Checkpoint: 0
3 [2012-05-21 16.40.19] PLANE_KIND "ForwardPlaneRight"
4 [2012-05-21 16.40.21] RES "NO ANSW " 2.2091629505157471
5 [2012-05-21 16.40.21] Status Correct: 0 - Wrong: 1 - Continuos correct: 0 - Checkpoint: 0
6 ...
7 [2012-05-21 16.41.19] PLANE_KIND "ForwardPlaneRight"
8 [2012-05-21 16.41.22] RES "CORRECT " 1.9394090175628662
9 [2012-05-21 16.41.22] Status Correct: 1 - Wrong: 1 - Continuos correct: 1 - Checkpoint: 0
10 [2012-05-21 16.41.25] PLANE_KIND "ForwardPlaneLeft"
11 ...
12 [2012-05-21 16.42.00] Status Correct: 10 - Wrong: 0 - Continuos correct: 0 - Checkpoint: 1
13 [2012-05-21 16.42.03] PLANE_KIND "ForwardPlaneLeft"
14 [2012-05-21 16.42.04] RES "CORRECT " 0.79603409767150879
15 [2012-05-21 16.42.04] Status Correct: 11 - Wrong: 0 - Continuos correct: 1 - Checkpoint: 1
16 [2012-05-21 16.42.07] PLANE_KIND "InversePlaneLeft"
17 [2012-05-21 16.42.09] RES "WRONG " 1.0926330089569092
18 ...
19 [2012-05-21 16.44.23] Status Correct: 42 - Wrong: 1 - Continuos correct: 29 - Checkpoint: 1
20 [2012-05-21 16.44.26] PLANE_KIND "ForwardPlaneLeft"
21 [2012-05-21 16.44.27] RES "CORRECT " 0.79077792167663574
22 [2012-05-21 16.44.27] Status Correct: 43 - Wrong: 0 - Continuos correct: 0 - Checkpoint: 2
23 ...

```

Listado 3.1: Extracto del archivo de estadísticas de uso del juego Avioncito

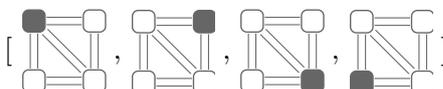
cada línea de dicho registro:

1. Fecha y hora de inicio del juego
2. Estado general del juego
3. Mostrando un avión amarillo para la derecha
4. Resultado: No respondió
- ⋮
8. Resultado: respondió correctamente en 1.93 segundos
- ⋮
10. Mostrando un avión amarillo para la izquierda
- ⋮
12. Estado del juego: ya pasó el *checkpoint* 1
- ⋮
16. Mostrando un avión rojo para la izquierda
17. Respuesta incorrecta
- ⋮
22. Estado del juego: en el *checkpoint* 2

En el caso particular de Casitas, además de los registros antes mencionados, al inicio de cada ensayo se menciona la posición de las casas y luego la posición de los personajes. Entre posición y posición, se registran los siguientes eventos del *mouse*: que el niño hizo clic sobre un personaje y mantiene apretado, que suelta el botón. Luego

se registra el resultado de soltar el botón: puede ser que el personaje seleccionado no se mueva porque fue soltado fuera de una casa o en su punto de origen, o que sí se mueva, cambiando la configuración de los personajes sobre el tablero.

La interpretación de la línea donde se detalla la posición de las casas es la siguiente:



y en cada una de las posiciones se ubica un número: 0 para indicar que está vacía, 1 para la casa de Ana, 2 para la casa de Pancho y 3 para la casa de Nubis.

La interpretación de la línea que detalla la posición de los personajes se interpreta de la siguiente manera:

[#movidas ya realizadas, posición de Ana, posición de Pancho, posición de Nubis]

y para las posiciones, 0 es la posición de arriba a la izquierda, 1 para arriba a la derecha, 2 para abajo a la derecha y 3 para abajo a la izquierda.

Se puede observar en detalle un extracto del archivo en el cual se guarda el registro de estadísticas de uso en el Listado 3.2. A continuación se detalla lo expresado en

```

1 [2012-05-16 10.18.23] INIT INICIANDO JUEGO
2 [2012-05-16 10.18.23] NEW *****
3 [2012-05-16 10.18.23] HOUSES [1, 2, 3, 0]
4 [2012-05-16 10.18.23] PLAYER [0, 3, 0, 2]
5 [2012-05-16 10.18.37] MOUSE DOWN 1
6 [2012-05-16 10.18.40] MOUSE UP
7 [2012-05-16 10.18.40] NOT MOVING 1
8 [2012-05-16 10.18.42] MOUSE DOWN 2
9 [2012-05-16 10.18.49] MOUSE UP
10 [2012-05-16 10.18.49] PLAYER [1, 3, 1, 2]
11 [2012-05-16 10.18.49] MOUSE DOWN 2
12 [2012-05-16 10.18.49] MOUSE UP
13 [2012-05-16 10.18.49] NOT MOVING 2
14 [2012-05-16 10.19.02] MOUSE DOWN 1
15 [2012-05-16 10.19.10] MOUSE UP
16 [2012-05-16 10.19.10] PLAYER [2, 0, 1, 2]
17 [2012-05-16 10.19.10] WON IN 2
18 [2012-05-16 10.19.10] STATUS Level: 1, Correct: 1, Solved_in_checkpoint: 1, Wrong: 0, Kind: 1
19 ...
20 [2012-05-14 18.13.15] LOST IN 2

```

Listado 3.2: Extracto del archivo de estadísticas de uso del juego Casitas

cada línea de dicho registro:

1. Fecha y hora de inicio del juego
2. Inicio de un ensayo
3. Posición de las casas en el tablero
4. Estado del ensayo, valor inicial
5. *Mouse* clickeado sosteniendo al personaje 1
6. *Mouse* liberado
7. Resultado de la movida: no se movió
- ⋮
10. Estado del ensayo después de mover
- ⋮
17. Trial ganado en 2 movidas

18. Estado general del juego

⋮

20. Juego perdido

Por último, Memomarote, en el que además de los registros comunes a los juegos, se agrega, al inicio de cada ensayo, la ubicación de las cajas donde aparecerán las fichas. Luego está el registro de qué fichas se agregan y como es que están construidas (forma, fondo, color, si tiene baldes, etc.). Después se pasa a registros del juego, se informa para cada clic si fue correcto (OK) o si fue erróneo (WRONG). En caso de ser correcto, se registran también las nuevas posiciones de las fichas.

Se puede observar en detalle un extracto del archivo en el cual se guarda el registro de estadísticas de uso en el Listado 3.3. A continuación se detalla lo expresado en

```

1 [2012-05-29 01.56.44] Add BOX (523, 620)
2 ...
3 [2012-05-29 01.56.44] Add BOX (783, 230)
4 [2012-05-29 01.56.44] Add CARD {pos: (523, 620), shape: 7, back: 4, props: [(1, Star, 2),]}
5 [2012-05-29 01.56.44] Add CARD {pos: (263, 620), shape: 2, back: 7, props: [(1, Star, 2),]}
6 [2012-05-29 01.56.44] Add CARD {pos: (523, 360), shape: 11, back: 2, props: [(1, Star, 8),]}
7 [2012-05-29 01.56.48] Card {pos: (263, 620), shape: 2, back: 7, props: [(1, Star, 2),]}
8 [2012-05-29 01.56.48] Card clicked OK
9 [2012-05-29 01.56.48] Move CARD from (523, 620) to (653, 360)
10 ...
11 [2012-05-29 01.56.48] Move CARD from (523, 360) to (653, 620)
12 ...
13 [2012-05-29 01.57.08] Card clicked OK
14 [2012-05-29 01.57.08] Game WON
15 [2012-05-29 01.57.09] Status Correct: 1 - Wrong: 2 - Continuos correct: 1 - Checkpoint: 1
16 ...
17 [2012-05-29 01.57.22] Card {pos: (523, 360), shape: 6, back: 9, props: [(1, Star, 4),]}
18 [2012-05-29 01.57.22] Card clicked WRONG
19 [2012-05-29 01.57.23] Status Correct: 1 - Wrong: 3 - Continuos correct: 0 - Checkpoint: 1

```

Listado 3.3: Extracto del archivo de estadísticas de uso del juego Memomarote

cada línea de dicho registro:

1. Agregando una caja en la posición (523, 620)
- ⋮
3. Agregando una caja en la posición (783, 230)
4. Agregando ficha
5. Agregando ficha
6. Agregando ficha
7. Ficha cliqueada
8. Resultado parcial
9. Moviendo ficha de posición
- ⋮
11. Moviendo ficha de posición
- ⋮
13. Resultado parcial
14. Fin del ensayo, se cliquearon todas las fichas una vez
15. Estado del juego
- ⋮
17. Ficha cliqueada
18. Resultado parcial, fin del ensayo
19. Estado del juego

El detalle de las propiedades de las fichas puede verse en la Tabla 3.1.

Propiedad	Valor
Posición	es un par (X, Y) con, $X \in [133, 263, 393, 523, 653, 783]$ $Y \in [230, 360, 490, 620]$
Forma (<i>shape</i>)	
Fondo (<i>back</i>)	
Sombrillas	
Estrellas	
Baldes	
Personajes	

Tab. 3.1: Propiedades de las fichas de Memomarote

3.2.2. Implementación del servidor central

Como fue indicado anteriormente, el cliente se conecta con un servidor central. El mismo fue implementado con las siguientes características:

- Se implementó el servidor central utilizando tecnologías libres: dentro de la oferta de *software* disponible se priorizó las que son más familiares al grupo de trabajo, lo cual llevó a que se utilizara de sistema operativo Ubuntu Server, en su versión estable más reciente (12.04 LTS); como base de datos, Postgresql 9.1, cuya descripción se puede leer en 3.2.3; como servicio web el *framework* CherryPy [31] que está desarrollado en Python.

Este esquema fue pensado para poder distribuir tanto la carga entre varios servidores por DNS (*Domain Name System*) como la posibilidad de instalar servidores intermedios e incluso armar una red en una zona sin Internet y lograr su funcionamiento de igual modo.

El servicio web implementado con CherryPy recibe todos los pedidos de los clientes y los procesa realizando las consultas a la base de datos que necesite dependiendo del caso.

- Para el desarrollo y posterior puesta en producción del servidor, se contó con dos equipos iguales, uno configurado como servidor de producción (<http://grot.matemarote.com.ar/>) y el otro como servidor de desarrollo (<http://asad.exp.dc.uba.ar:6345/>): Intel(R) Xeon(TM) CPU 2.80GHz con 16GB de memoria RAM. Ambas máquinas se encuentran instaladas en la sala de servidores del Departamento de

Computación [32] de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires.

- Para poder cumplir con el requerimiento de tener separados los distintos tipos de usuarios, se implementó un sistema de perfiles. Cada perfil corresponderá a un programa institucional, como puede ser OLPC La Rioja, u OLPC Uruguay. Cada perfil tendrá asignadas las actividades y juegos en las versiones que correspondan según el programa.

De esta manera se cumple con la posibilidad de, sobre una misma infraestructura, tener personalizado el *framework* para cada niño dependiendo del programa al que pertenece.

Como no se contó con una lista de máquinas sobre las cuales se iba a trabajar se programó el servidor para que realice el registro automático de máquinas nuevas al recibir un pedido de inicio de sesión de una máquina nueva. Este registro implica además que a esa computadora se le asigna el perfil de OLPC La Rioja.

Se puede ver el detalle de la implementación de los perfiles en la base de datos en 3.2.3.

- Para verificar el correcto funcionamiento del servidor se implementaron 2 grupos de tests, uno para el acceso web y otro para la base de datos. Ambos grupos verifican las respuestas a todas las funcionalidades que ofrece el servidor. Están implementados utilizando PyUnit [33] y esta batería de pruebas fue corrida cada vez que se realizó una modificación en el servidor.
- Para tener registro en tiempo real de la actividad que está teniendo el servidor, en cuanto a pedidos y a localización estos, se utilizó *Google Analytics* [34] dentro de CherryPy. Ante cada consulta al servidor, éste realiza un llamado a *Google Analytics* para registrar el acceso.

3.2.3. Bases de datos

A continuación se presenta en detalle la estructura de las bases de datos involucradas en el proyecto. En primer lugar se muestra la base de datos en el cliente, para la cual se utilizó SQLite ya que es simple, liviano, está implementado sobre archivos y Python trae soporte para utilizar este motor. Luego la base de datos del servidor, cuyo motor es Postgresql 9.1.

Base de datos del *framework*

La base de datos del cliente, implementada con SQLite, presenta un esquema con 4 tablas que lo podemos dividir en 2 grupos: juegos y *framework* y los usuarios. Se puede ver el esquema en la figura 3.3. A modo de breve descripción, a continuación se listan las tablas agrupadas como se expresó anteriormente y se comenta su uso.

Tabla para los juegos:

Tabla games:

Aquí se almacenarán los juegos instalados, su versión, su estado, su ruta de ejecución, el orden que debe ocupar en la lista de aplicaciones, ícono, título, descripción, etc.

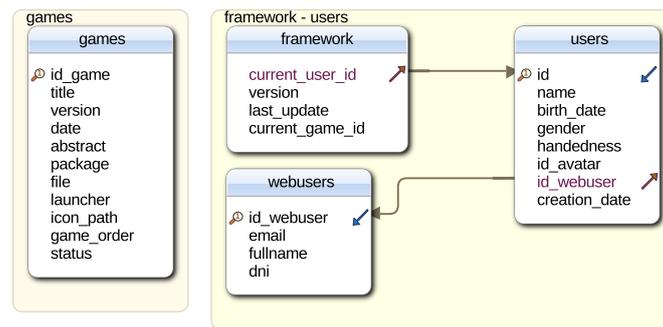


Fig. 3.3: Esquema de la base de datos del *framework*. La base está separada en dos grupos, uno que contiene toda la información referida a los juegos instalados (*games*) y otra del *framework* y los usuarios (*framework - users*).

Para el *framework* y los usuarios:

Tabla framework:

Mantiene el registro de cuál es el usuario activo, qué versión del *framework* se encuentra instalada y a qué perfil pertenece la computadora

Tabla users:

Almacena la información de los perfiles de los usuarios

Tabla webuser:

Guarda el identificador de webuser, su correo electrónico y su nombre

En resumen, este esquema soporta la creación de distintos usuarios en una misma máquina (tanto usuario local como usuario web), mantener el estado del *framework* y la información sobre los juegos.

Base de datos del servidor

La base de datos del servidor, implementada con Postgresql 9.1, presenta un esquema de 16 tablas que la podemos dividir en 5 grupos según su funcionalidad: los juegos, donde se guardan detalles y versiones de estos; las computadoras y perfiles, donde se almacenan las computadoras registradas, los perfiles y las versiones que tiene que tener de cada juego; el *framework*, que contiene las distintas versiones; los usuarios, donde se guardan los datos de los usuarios locales de cada máquina; y las estadísticas de uso y sesiones, donde se almacenan las estadísticas de uso enviadas desde las computadoras de los niños y los pedidos realizados al servidor. Se puede ver el esquema en la figura 3.4. A modo de breve descripción, a continuación se listan las tablas agrupadas como se expresó anteriormente y se comenta su uso.

Tablas para los juegos:

Tabla games:

Guarda la lista de juegos que existen en el sistema

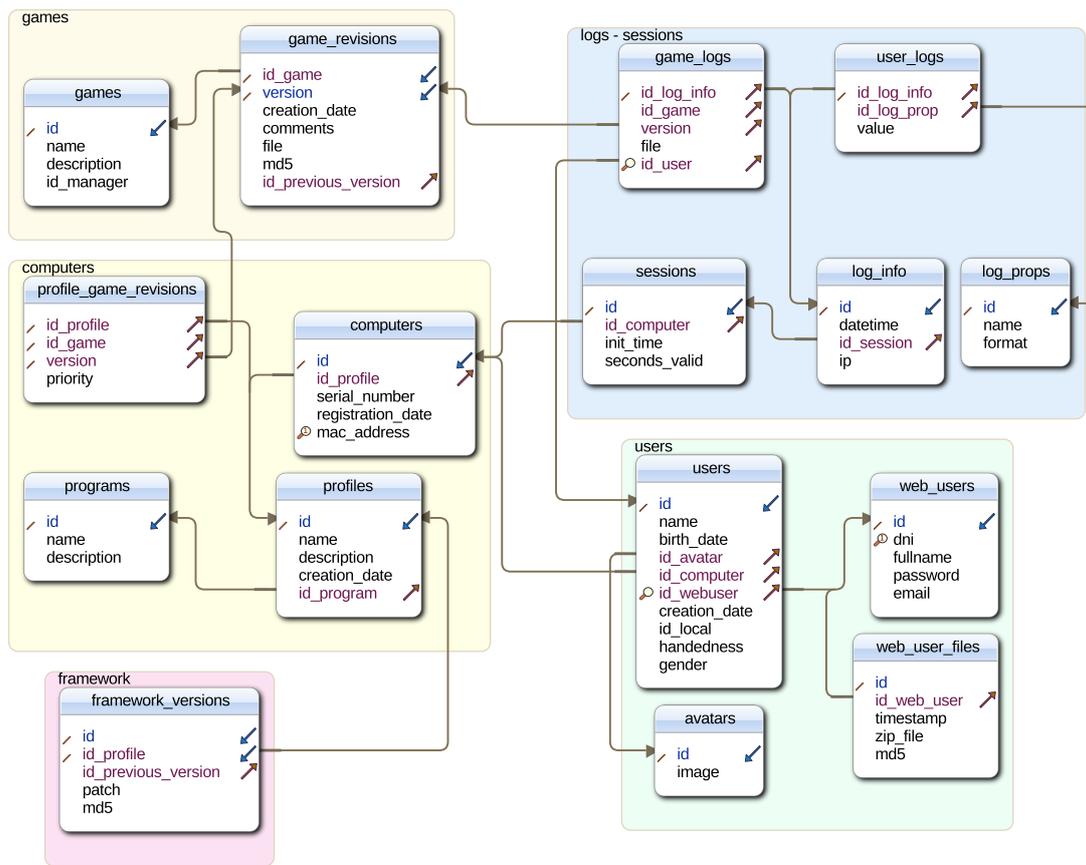


Fig. 3.4: Esquema de la base de datos del servidor. Se observan 5 grupos de tablas que se corresponden con la información guardada sobre juegos (**games**); estadísticas de uso y sesiones (**logs - sessions**); computadoras y perfiles de instalación (**computers**); usuarios (**users**); y versiones del *framework* (**framework**).

Tabla `game_revisions`:

Tiene la lista de versiones de una actividad, desde el instalador inicial hasta el archivo binario de cada actualización

Tablas para las computadoras y perfiles:

Tabla `computers`:

Acá se almacena la información necesaria para identificar una *netbook*, actualmente la *mac address* y el número de serie (sólo en el caso de las OLPCs)

Tabla `profiles`:

Esto permite definir distintos perfiles para un mismo programa (por ejemplo: Ceibal primaria, Ceibal secundaria)

Tabla programs:

Aquí se almacenarán los programas institucionales en los que el *framework* puede estar instalado (por ejemplo: Programa Joaquín V. González, ConectarIgualdad, Ceibal, etc.)

Tabla profile_game_revisions:

Establece el vínculo entre un perfil y una versión de una actividad, lo que indica cuál es la versión que debería tener dicho perfil instalada

Tabla para el *framework*:

Tabla framework_versions:

Aquí estará guardado el binario del *framework* (o su actualización), además de generar la estructura de orden entre las versiones

Tablas para los usuarios:

Tabla users:

Es el lugar donde estarán almacenados los datos que identifican al usuario, como puede ser: nombre de usuario, género, mano hábil, etc.

Tabla web_users:

Almacena los datos de un usuario de una máquina para permitir la migración a otra computadora

Tabla web_users_files:

Almacena los archivos de un *webuser*

Tabla avatars:

Listado de avatars posibles para los usuarios

Hasta acá todas las tablas corresponden a la parte que provee el software, lo que queda es la parte que recibe las estadísticas de uso y guarda registro de las conexiones con el servidor:

Tabla sessions:

Cada transacción web que no sea iniciar una sesión, tendrá como requisito tener una, la cual se asocia a una computadora registrada. Esta tabla contiene esta información

Tabla log_info:

Información general de los registros, como la fecha, sesión e ip

Tabla user_logs:

Almacena los eventos producidos por la interacción entre las *netbooks* y el servidor, como el inicio de sesión, chequeo de nuevas versiones de actividades, *downloads*, *uploads*, errores, etc.

Tabla game_logs:

Guarda las estadísticas de uso de las actividades, cada envío se guarda en un archivo de formato zip dentro del cual estará registrada la información correspondiente a dicha actividad en el formato preestablecido

Tabla log_props:

Guarda un listado de propiedades predefinidas para generar registros

En conclusión, este esquema permite proveer las distintas versiones de *framework* y los instaladores de los juegos en sus diferentes versiones, almacenar la lista de computadoras registradas, los usuarios creados en las *netbooks* y las estadísticas de uso de los juegos y del servidor.

4. RESULTADOS

Como resultado de esta tesis tenemos una herramienta general que permite realizar intervenciones: Mate Marote.

El sistema consta de dos partes, el servidor y el cliente.



Fig. 4.1: Servidores del proyecto. El de arriba es Bochini, el servidor de producción; y el de abajo es Asad, el de desarrollo

En la Figura 4.1 se muestra la configuración final de los servidores y en la Figura 4.2 se muestran capturas de pantalla del cliente.

En la parte del cliente, o *framework*, para la prueba piloto se incluyeron 3 juegos: Avioncito que es una tarea de flexibilidad cognitiva y control inhibitorio, Casitas que es una tarea de planeamiento, y Memomarote que se enfoca principalmente en la memoria de trabajo.

Se puede observar en la Figura 4.2a una captura de pantalla del *framework*, donde se ven los juegos instalados y un botón para abrir el formulario que permite crear un usuario.

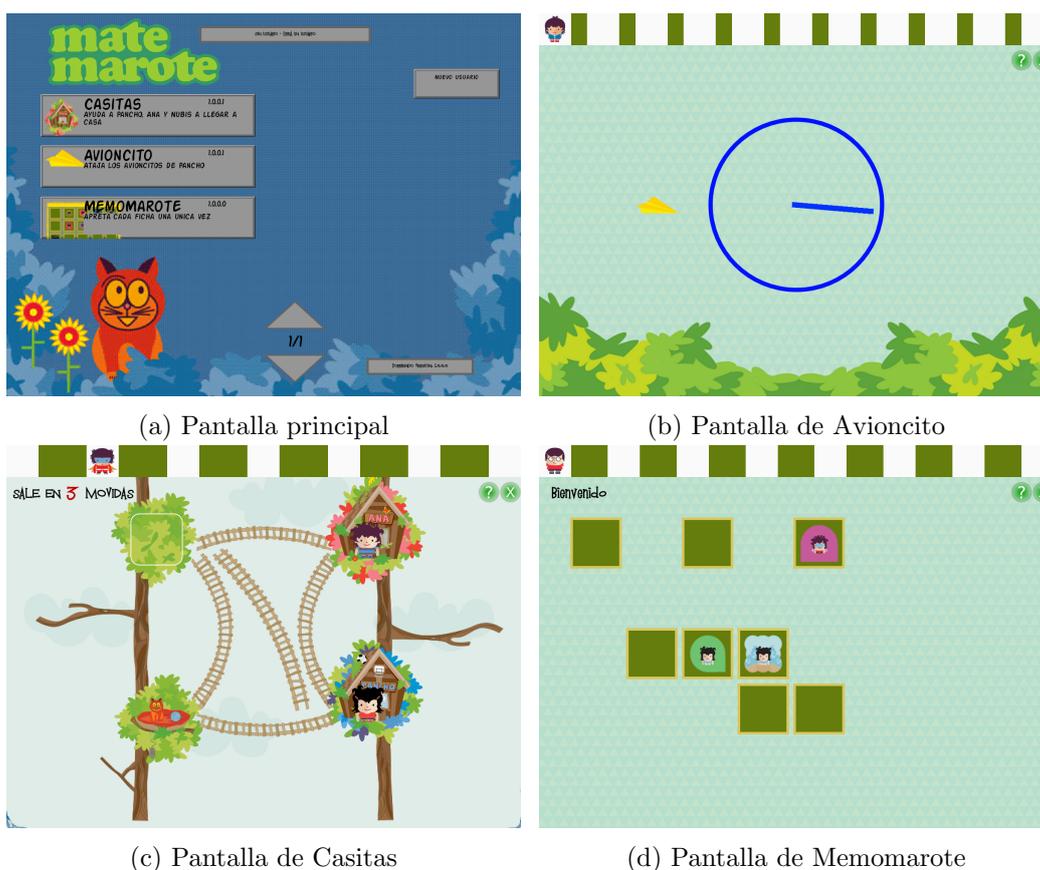
En la Figura 4.2b se observa una captura de pantalla del juego Avioncito en el cual se presenta como estímulo un avión amarillo que apunta a la derecha. Se puede ver en el centro de la imagen el reloj que señala el tiempo restante para responder el ensayo, que es exactamente una vuelta. Y arriba se ve el avance del juego.

En la Figura 4.2c se observa una captura de pantalla del juego Casitas. El ensayo presentado sale en tres movidas, como lo indica el texto presente en la pantalla, una solución óptima es: (1) mover a Ana por la diagonal a la casilla de arriba a la izquierda, (2) mover a Pancho a su casa, (3) mover a Ana a su casa. Se puede ver la misma barra de nivel que en Avioncito, pero el personaje se encuentra en el segundo casillero, indicando que se encuentra en el segundo nivel (*checkpoint*).

En la figura 4.2d se observa una captura de pantalla del juego Memomarote. Se observan 8 casillas de las cuales 3 tienen una ficha en ella: hay una ficha verde, con forma de gota y tiene a Ana; hay otra con forma de escudo, de fondo violeta y que tiene a Pancho; por último una que tiene una forma compuesta por varios círculos, que el fondo es una playa y en la que nuevamente está Ana.

Todos los juegos tienen en común la barra de niveles, cuyo largo varía de un juego a otro, y dos botones en la parte superior derecha de la pantalla: uno con un ? que, al ser presionado, muestra un cartel con las instrucciones, y el otro con una X que sirve para salir del juego.

La primera prueba piloto fue realizada en La Rioja. El lugar elegido fue propuesto por el Dr. Antonio Battro, que es *Chief Education Officer* de OLPC, que trabajó en conjunto con el Ministerio de Educación de La Rioja para poner en funcionamiento allí al Programa

Fig. 4.2: Capturas de pantalla del *framework*

Joaquín V. González y concretar la llegada y entrega de unas 60.000 XOs para los niños y docentes Riojanos. El Dr. Battro, a su vez, trabaja en colaboración con varios integrantes del Laboratorio de Neurociencia Integrativa [35], que es dónde nace originariamente este proyecto. Con lo cual, a través del Dr. Battro, se estableció contacto con las autoridades y se planificó realizar una prueba piloto a fin de evaluar el funcionamiento de esta herramienta en el lugar donde, de ser exitosa la prueba, se implementaría a gran escala.

La prueba piloto de esta nueva herramienta consistió en viajar a La Rioja y realizar la presentación del proyecto Mate Marote así como su implementación particular para La Rioja a los referentes del Programa Joaquín V. González. Luego, se procedió a instalar el *framework* en más de 100 XOs de escuelas seleccionadas, donde se capacitó a docentes y directivos sobre el uso del mismo.

4.1. Prueba piloto: La Rioja

Para realizar la primera prueba piloto había que instalar el *framework* en máquinas de niños, por lo cual se realizó un viaje a la ciudad de La Rioja del día 15 al 17 de mayo de 2012. La comitiva estuvo compuesta por el Dr. Antonio Battro, el Dr. Diego Fernández Slezak, la Lic. Andrea Goldin y el autor de esta tesis, Matías López y Rosenfeld. En el primer día del viaje hubo una charla técnica donde se presentó el *framework*, se comentaron antecedentes y objetivos del programa en la sede de la Unidad Provincial de Tecnología

de la Información y la Comunicación (UPTIC). Durante el segundo día se recorrieron 4 escuelas, dando capacitación a docentes de los establecimientos e instalando las *netbooks* de los alumnos de primer y segundo grado de dichas escuelas. Se instalaron alrededor de 100 computadoras. Para finalizar, el último día, se dio a los docentes una charla de divulgación en el Instituto Superior de Formación Docente “Albino Sánchez Barros”. Esta visita fue anunciada en medios locales de La Rioja [36].

Los primeros resultados de las instalaciones del *framework* en las OLPC de La Rioja mostraron que el rendimiento gráfico de las computadoras de los niños es inferior al esperado. Los juegos fueron testeados en el Departamento de Computación de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires sobre el mismo modelo de XOs de La Rioja. Sin embargo, las XOs donde fue testado el *framework* antes del viaje eran máquinas recién instaladas, sin ningún tipo de *software* agregado al que viene de fábrica. Por el contrario, las máquinas de los niños –casi en su totalidad– tenían el disco lleno de actividades, lo que provocaba que el sistema en general anduviera más lento que en las máquinas donde fue probado. El mayor deterioro de rendimiento fue observado en el juego Avioncito, donde lo que se busca es que el niño responda lo más rápido posible al estímulo, y la computadora no respondía como lo esperado.

Al volver a Buenos Aires, se procedió a optimizar al máximo el rendimiento de las aplicaciones afectadas debido a la sobrecarga de actividades de las XOs de los niños. Esta modificación dio a lugar a la primera actualización del sistema a gran escala. Todos los niños que iniciaran el *framework* verían la actualización de Avioncito. Este procedimiento resultó exitoso ya que prácticamente no se reportaron errores en la actualización.

También se arregló una falla en el juego Casitas que provocaba que no se guardara el avance entre una sesión de juego y otra. Al igual que la anterior, fue subida al servidor y se actualizaron las XOs exitosamente.

Por otro lado, el equipo de técnico del programa Joaquín V. González (llamados Facilitadores TICs), se encontraba terminando la imagen para las XOs 2013, y muchas de las máquinas que entraron a reparación o fueron recientemente repartidas se encontraban instaladas con una imagen *beta* del sistema. Estas *netbooks* instaladas con la imagen nueva no andaban de forma correcta, mostrando severos *bugs* que muchas veces imposibilitaban el acceso al *framework*. Por ejemplo, uno de los errores mas predominantes fue que el sistema operativo capturaba el teclado y no dejaba que el *framework* lo registrara, impidiendo que el niño pudiera interactuar con los juegos. Se prevee viajar a La Rioja a fin del año 2012 o comienzos del 2013 para realizar las verificaciones sobre la nueva imagen previamente al inicio del ciclo escolar 2013.

Una cosa descubierta durante el viaje es que muchos educadores de escuelas rurales (que aún no poseen punto de acceso a Internet) actualizan su computadora durante algún viaje a zonas con conectividad, para luego compartir las actualizaciones en su escuela de origen, no siendo estrictamente necesario que cada máquina posea una conexión a Internet para ciertas mejoras en el sistema. Estas máquinas, mientras no se conecten a Internet no nos permiten conocer cuál fue su uso, si el *framework* es exitoso, ni recuperar las estadísticas de uso.

4.1.1. Monitoreo de actividad

Al regresar del viaje comenzó la tarea de monitoreo de uso del *framework*. Para medir la actividad de uso del *framework*, se decidió contar la cantidad de archivos con registros de estadísticas de uso recibidos de cada máquina. Esta medida aglutina dos aspectos del

monitoreo: 1) acceso a conexión a Internet y 2) uso de los juegos y estadísticas de uso. Si se detecta uno de estos archivos, significa que el niño jugó al menos una vez desde su último envío de estadísticas, y este registro fue subido al servidor con conexión a Internet.

El viaje para la prueba piloto en La Rioja fue el 15 de Mayo de 2012. Ese mismo día se instalaron aproximadamente 20 máquinas de docentes y, al día siguiente, más de 100 computadoras de alumnos. Luego, el 28 de Mayo, se subieron al servidor de producción (disponibles para las máquinas de La Rioja cuando tuvieran conexión) las 2 versiones que corregían errores detectados durante la visita. Teniendo en cuenta estas fechas, se consultó la actividad: el 16 de Mayo, en pleno viaje; el 28 de Mayo, día en que se subieron las 2 versiones nuevas de Avioncito y Casitas; y, luego mensualmente. Los resultados pueden verse en la tabla 4.1.

Inicialmente, el día de la instalación se observan unos pocos registros de estadísticas de uso, alrededor de 50 archivos recibidos, que corresponden a las capacitaciones docentes realizadas durante la instalación. Como es de esperar, se observa un pico inicial de nuevos registros en los primeros 15 días, para luego caer a medida que avanza el tiempo. Sin embargo es destacable notar que la actividad no cesa nunca, incluso más de 3 meses después del viaje. Es interesante ver que, pese a que existe una versión nueva de algunos juegos (Casitas 1.0.0.1 y Avioncito 1.0.0.1), se siguen recibiendo estadísticas de uso de los juegos en su primera versión, indicando que dichas computadoras no tuvieron conexión hasta el momento pero el *framework* fue abierto y los niños jugaron, por lo cual recibimos sus estadísticas de uso de las primeras versiones.

Juego	Versión	Cantidad de <i>uploads</i> por período						
		16-05	28-05	28-06	28-07	28-08	28-09	6-11
Casitas	1.0.0.0	26	156	52	18	13	9	27
Casitas	1.0.0.1	-	-	22	12	15	8	26
Avioncito	1.0.0.0	20	120	48	14	10	7	23
Avioncito	1.0.0.1	-	-	19	7	11	5	18
Memomarote	1.0.0.0	13	110	55	21	20	14	39

Tab. 4.1: Cantidad de archivos recibidos con estadísticas de uso entre el 16 de mayo de 2012 y el 6 de noviembre de 2012

Cada uno de estos *uploads* incluyen registros de estadísticas de uso de los juegos que se utilizaron en cada una de las máquinas, encapsulados en un archivo *.zip*. A continuación, en la sección 4.1.2, se presenta un análisis de las estadísticas de uso, juego por juego, a partir de los datos recolectados.

Por otro lado, podemos hacer un análisis sobre los pedidos al servidor. Desde el 15 de mayo al 6 de noviembre tenemos más de 2000 pedidos para chequear si había una nueva versión de un juego respondidos; casi 1600 inicios de sesión; más de 900 archivos recibidos por el servidor; más de 700 verificaciones de que el *framework* esté en la versión correcta; más de 700 pedidos de la lista juegos instalados; más de 350 actualizaciones a la versión 1.0.0.1 (actualización de un juego ya instalado); casi 100 envíos de usuarios; y 85 instalaciones de juegos. Este resumen de la actividad se puede ver en la Tabla 4.2.

Por último, analizamos la cantidad de nuevas XOs registradas en períodos de tiempo, podemos ver que siguieron apareciendo nuevas máquinas, lo cual se explica porque el *framework* es compartido entre los niños. Por esta razón es que, si bien inicialmente se instalaron 100 computadoras, el número de equipos asciende ahora a las 300. Se puede

Tipo de pedido	Cantidad de pedidos atendidos
Verificar versión de un juego	2136
Iniciar sesión	1598
Recibir archivo con estadísticas de uso	927
Verificar versión del <i>framework</i>	781
Pedir lista de juegos	770
Enviando nueva versión	362
Recibiendo datos de usuarios	97
Enviando juegos nuevos para instalar	85

Tab. 4.2: Cantidad de pedidos que respondió el servidor

observar el detalle de las máquinas nuevas en la Tabla 4.3

Período de tiempo	Cantidad de máquinas nuevas
15-05 al 15-06	210
15-06 al 15-07	20
15-07 al 15-08	24
15-08 al 15-09	1
15-09 al 15-10	36
15-10 al 06-11	9
Total	300

Tab. 4.3: Cantidad de máquinas nuevas registradas en el servidor por mes

4.1.2. Análisis de estadísticas de uso

Los registros generados a partir del uso de los juegos por parte de los niños están diseñados para poder extraer marcadores estereotipados sobre el comportamiento frente a los juegos. Cada *upload* contiene los datos de un juego, pudiendo incluir varias sesiones de uso. A continuación se presentan análisis sobre el uso de cada juego por separado (Avioncito, Casitas y Memomarote).

Avioncito

Como se describe en la sección 2.1.1, este juego compete a la capacidad cognitiva del control inhibitorio, y consiste en que el jugador debe indicar la dirección en la que se lanzan avioncitos de papel. En caso de que el avión sea amarillo, debe indicarse la dirección destino. En caso de un avión rojo, debe indicarse la dirección opuesta.

Intuitivamente, se espera que la respuesta a un avión amarillo sea más fácil que la de un avión rojo, ya que este último requiere una operación adicional al primer caso, i.e. la negación del estímulo recibido [20]. Es posible medir esta diferencia en la dificultad en términos del tiempo de respuesta del niño. A mayor dificultad, más lento responde.

Para verificar esta hipótesis, se analizaron los tiempos de respuesta registrados en las estadísticas de uso, únicamente sobre los ensayos donde el niño respondió correctamente, discriminando los tiempos provenientes de un color de avión y de otro. Es decir que tenemos 4 opciones de respuesta: avión amarillo mirando en dos direcciones, izquierda

(AI) y derecha (AD); avión rojo mirando en dos direcciones, izquierda (RI) y derecha (RD).

Para realizar este análisis se consideraron los tiempos de respuesta para cada tipo de avión. La cantidad de ensayos de cada tipo se encuentra detallado en la tabla 4.4.

Imagen	Nombre	Cantidad de ensayos
	AD	988
	AI	948
	RD	239
	RI	236

Tab. 4.4: Cantidad de ensayos por tipo de avión

En la Figura 4.3 se muestra el tiempo de respuesta únicamente sobre las respuestas correctas. Se observa que el tiempo para aviones de color rojo es mayor al de los amarillos.

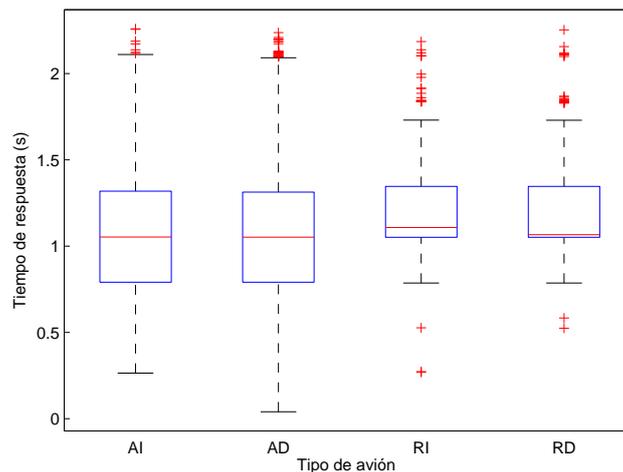


Fig. 4.3: Tiempo de respuesta en segundos por tipo de avión: aviones amarillos mirando a la izquierda (AI); aviones amarillos mirando a la derecha (AD); aviones rojos mirando a la izquierda (RI); aviones rojos mirando a la derecha (RD).

Esta diferencia resulta significativa tanto para AI y RI (t -test, $p = 1 \times 10^{-06}$) como para AD y RD (t -test, $p = 1 \times 10^{-05}$). Por otro lado, vemos que entre aviones del mismo color la diferencia no es significativa: RD-RI (t -test, $p = 0,63$) y AD-AI (t -test, $p = 0,62$).

Casitas

Como se describe en la sección 2.1.2, este juego requiere que el niño planifique una serie de pasos con el fin de lograr un objetivo. Se juega en un tablero que tiene la forma de un cuadrado, donde los personajes se ubican en los vértices y hay caminos que los unen en las aristas y un camino en una diagonal (ver Figura 2.1). La actividad consiste en llevar a los tres personajes participantes desde una configuración inicial a una final con algunas restricciones: no puede haber dos personajes en el mismo vértice; sólo pueden estar en los

vértices del tablero; se pueden mover de una posición a otra sólo si esta está vacía y hay camino que las una; y se mueven de a uno a la vez.

Las posibles configuraciones de tableros pueden verse en la figura 2.2. Cada ensayo del juego consiste en pasar de una configuración a otra. En la figura se puede apreciar que la distancia máxima entre dos configuraciones es de 7 pasos, y sin bien la mínima es 1 (ó 0 si se inicia desde la misma que la final), sólo se jugó con tableros entre 2 y 7.

En este contexto, esperamos ver que a medida que aumenta la distancia el juego resulta más difícil, lo cual se ve reflejado en la tasa de éxito para cada distancia. Por la naturaleza del problema, partiendo de la configuración que requiere como mínimo 7 movidas, cualquier decisión que tome el niño lo dejará a distancia 6; de ahí en más, si no retrocede, estará en 5. La mayoría de estas posiciones sólo permiten un movimiento que no implique retroceder, el cual ya los deja en 4. Con los tableros que salen en como mínimo 6 movidas, también debería observarse algo similar, salvo porque inicialmente existe la posibilidad de alejarse.

Los datos recolectados, que se pueden observar en la figura 4.4 confirman esta hipótesis: los tableros iniciales, distancia 2 y 3, son resueltos en la cantidad mínima de movidas en un porcentaje mayor al 90 %; el rendimiento cae un poco para distancia 4, aunque se mantiene sobre el 70 %; en 5 se observa el peor rendimiento (no alcanza el 30 %) y de ahí en más, como se esperaba ver, sube hasta superar el 70 % en el caso del tablero de distancia 7.

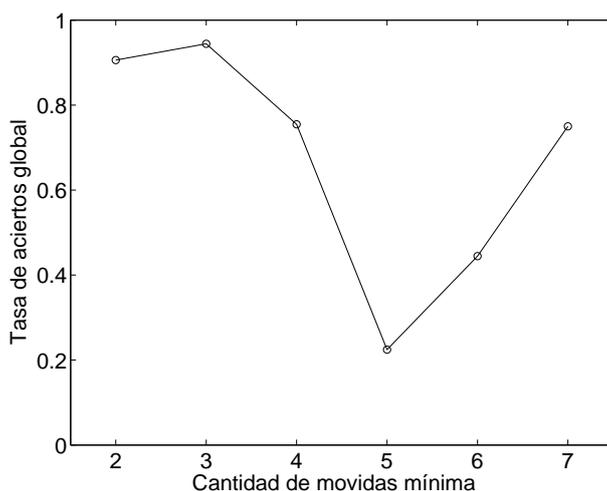


Fig. 4.4: Tasa de aciertos en los tableros de distinta distancia

Memomarote

Como se describe en la sección 2.1.3, este juego requiere que el niño recuerde una serie de fichas para lograr un objetivo. Al iniciar, el juego muestra una cantidad definida de fichas ubicadas en un cierto lugar del tablero. El jugador debe seleccionar una única vez cada una de las fichas, haciendo clic sobre ellas. Al cliquear una, las fichas desaparecen y reaparecen en una ubicación diferente. A medida que el jugador selecciona correctamente, sube su nivel aumentando la cantidad de fichas presentadas.

Trivialmente, el primer clic siempre es correcto, ya que no hay ninguna ficha seleccionada previamente. A medida que el jugador va seleccionando fichas, el ensayo se dificulta ya que las opciones de selección se ven reducidas, por lo que el jugador se ve obligado a generar alguna estrategia para ganar que puede ser recordar las selecciones previas, recor-

dar las flatantes, ordenar de algún modo (por ejemplo contando los baldes), etc. Uno de los objetivos de este juego, es detectar cuáles son los *atributos* utilizados para recordar las fichas.

En nuestro caso de estudio, el juego comenzó con tres fichas. Luego de instalado, se obtuvo el registro de juego por parte de los niños. El máximo número de fichas alcanzado por más de 5 niños fue 6.

En primer lugar, nos interesa dilucidar la calidad de los jugadores, teniendo en cuenta que este es un escenario no supervisado en donde los niños juegan sin un tutor a su lado. Para eso, nos ubicamos en cada uno de los niveles alcanzados (3 a 6 fichas) y medimos la cantidad de aciertos en cada uno de los clics del ensayo. En el panel izquierdo de la figura 4.5, se observa el porcentaje de aciertos promediado entre los jugadores (y su error estándar) para cada uno de los niveles analizados (azul: nivel 3, rojo: nivel 4, verde: nivel 5 negro: nivel 6). Como era de esperar, se observa un descenso de la tasa de aciertos a medida que avanzamos en el número de clic de un ensayo.

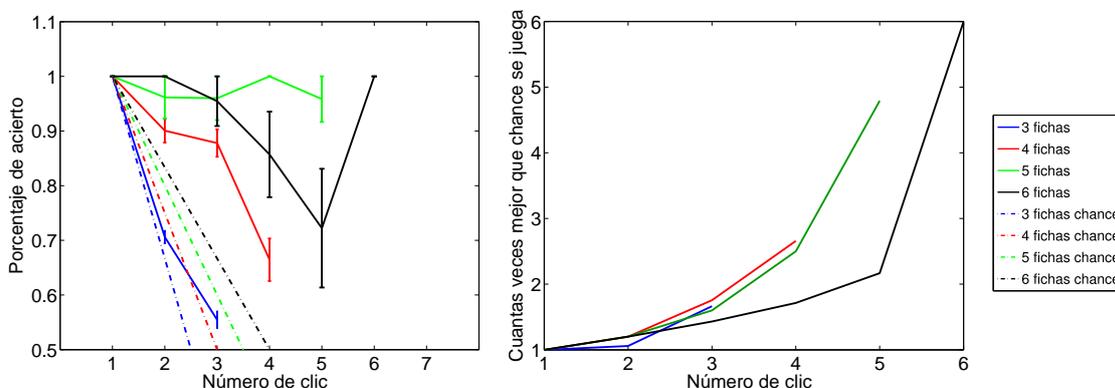


Fig. 4.5: Rendimiento en comparación con jugar al azar

Luego, se analizaron los datos recolectados para estudiar cuán bien jugaban los niños. Para ello, se calculó cuál sería la tasa de acierto para un jugador que elige fichas al azar (Figura 4.5, líneas punteadas). A partir de esta tasa de aciertos, se calculó la tasa de mejora con respecto a un jugador que selecciona al azar (Figura 4.5, panel derecho). Un valor de 1 implica que el jugador humano juega exactamente al mismo nivel que el agente aleatorio. Valores superiores a 1 indican cuánto mejor que un jugador aleatorio juega el niño (por ejemplo, al tercer clic en un ensayo de 4 fichas el niño acierta casi el doble de veces que si jugara al azar).

Como era de esperar, el agente azaroso se comporta igual que los humanos para el primer clic, ya que todas las selecciones son correctas. A medida que avanza el clic (y el nivel) los resultados obtenidos por los jugadores son mejores que los obtenidos por el agente. Es interesante notar que, a mayor nivel de juego, menor es la mejora obtenida con respecto al agente, indicado por la curva correspondiente al nivel 6 (negro) por debajo de las curvas de los niveles 4 (rojo) y 5 (verde). Por ejemplo, en el clic 5 el valor de la curva de nivel 6 indica que el jugador está seleccionando dos veces mejor que un agente aleatorio; por otro lado, el valor de la curva de nivel 5 llega a marcar entre 4 y 5 veces el valor de un agente.

Estos resultados nos indican que, para un mismo número de clic, la dificultad depende del nivel, es decir que depende de la cantidad de fichas totales del ensayo.

5. CONCLUSIONES

Una nueva disciplina, llamada neurociencia educacional, combina métodos de investigación en biología, ciencias cognitivas, del desarrollo y de la educación, para fortalecer y mejorar algunos aspectos del aprendizaje y la enseñanza.

La Argentina está viviendo una oportunidad única al tener una fracción significativa de los estudiantes y docentes que posee sus propias computadoras, siendo un fenómeno social que posiblemente este produciendo un cambio revolucionario en la educación. Estas computadoras son equipos de bajo costo, sin que esto implique que no cubren las necesidades para las que fueron pensados, que logran ser económicos por ser comprados en lotes de gran cantidad de unidades. Esto lleva a que se le entregue el mismo modelo de computadora a todos los niños.

La existencia de esta plataforma única de computadoras para alumnos abre la puerta a que algunos métodos de la neurociencia desembarquen en el mundo educativo realizando intervenciones educativas a nivel masivo.

En esta tesis se desarrolló una herramienta que explota esta nueva realidad: Mate Marote. Esta plataforma contiene aplicaciones/juegos que intervienen en el aprendizaje de los niños, potenciando capacidades cognitivas. Además, fue diseñada para que –una vez instalada en las máquinas– al encontrarse con conexión a Internet, se actualice tanto el *framework* como las aplicaciones que contenga. Por otro lado, enviará el registro de las estadísticas de uso al servidor central para su posterior análisis. Este análisis de los datos recolectados, producto del pensamiento humano, permitirá descubrir nuevos marcadores cognitivos sobre los cuales desarrollar nuevas intervenciones en forma de aplicación o juego del *framework*, que potencien alguna nueva capacidad cognitiva que a su vez registre estadísticas de uso, que nuevamente permitan reconocer nuevos marcadores. De esta forma se cierra un ciclo que todo el tiempo se está realimentando y generando nuevos contenidos.

Este *framework* fue puesto a prueba en la instalación realizada en La Rioja entre los días 15 y 17 de Mayo de 2012, en la cual se instalaron alrededor de 100 máquinas. Como resultado de este viaje verificamos que el *framework* podía ser instalado sin problemas en las computadoras de la imagen oficial del programa Joaquín V. González de La Rioja. Además, la conexión entre el *framework* y el servidor central fue exitosa, tanto para la actualización del *framework* como para la actualización de las 3 aplicaciones incluidas en el *framework*: Avioncito, Casitas y Memomarote y, a su vez, en el servidor central se recibieron archivos con las estadísticas de uso de todos estos juegos en sus distintas versiones.

A partir de estas estadísticas de uso recibidas se pudo hacer un seguimiento de actividad de las computadoras, comprobando que, incluso bastante tiempo después del viaje, la actividad en las distintas máquinas instaladas continuó y, si bien bajó, no se extinguió definitivamente. Además, se registró un gran número de computadoras nuevas producto de que los niños comparten las aplicaciones que les gustan entre ellos.

Por otro lado, los análisis realizados sobre las estadísticas de uso son consistentes con los resultados obtenidos en las experiencias previas en las escuelas. Sin embargo, estos nuevos datos fueron obtenidos sin supervisión; es decir, nadie estaba al lado de los niños mientras ellos utilizaban los juegos de Mate Marote. Este resultado sugiere que el proyecto es escalable a niveles masivos ya que, aun sin contar con instructores guiando la actividad,

se conservan las propiedades de la versión supervisada.

Esta prueba piloto sugiere que la disponibilidad de una plataforma digital estandarizada, que permite la obtención y el análisis de registros producto del pensamiento, posibilita el diseño de actividades para la mejora del proceso de enseñanza/aprendizaje. Así, la herramienta aquí presentada abre una ventana hacia nuevas estrategias de enseñanza a través de intervenciones educativas a nivel masivo.

6. TRABAJO A FUTURO

En primer lugar, deben realizarse ajustes pendientes sobre los juegos instalados durante la prueba piloto. En particular, el registro de estadísticas de uso de Avioncito no incluye la ubicación del avión mostrado, y la falta de este dato no permite identificar un sesgo en las respuestas relacionado a la posición de exposición. Además, a partir del rendimiento observado durante la prueba piloto en las computadoras sobrecargadas de los niños, debe optimizarse la velocidad de refresco de Avioncito para una mejor fluidez del movimiento de la aguja del reloj. Por otro lado, las reglas de generación de fichas de Memomarote son aleatorias. Para una mayor efectividad en el análisis de los registros de uso, deben definirse reglas generativas que permitan agrupar las fichas, facilitando el avance a niveles con mayor cantidad de elementos. En forma general, se estudiará la posibilidad de incorporar sonidos a los juegos ya existentes, que fue pospuesto debido al rendimiento de las XOs. Adicionalmente, teniendo en cuenta las edades de los niños involucrados inicialmente, se incorporarán videos explicativos sobre las reglas de los juegos para aquellos que no saben leer.

El éxito de un juegos puede medirse en la cantidad de tiempo que un usuario le dedica [28]. Para mejorar este índice, se incorporarán animaciones al juego para aumentar el entretenimiento de los niños. Por último, en esta misma línea se diseñará un sistema de puntos global para que los niños tengan una métrica propia que quieran superar.

Durante los cursos de capacitación, asistieron docentes de zonas rurales sin conexión que copiaron el paquete de instalación de Mate Marote para distribuirlo manualmente en las escuelas a su cargo. Los registros de uso de estas computadoras, probablemente, no serán enviados al servidor. Como posible solución a este problema, se encarará la instalación y configuración de una versión del servidor para usar en zonas remotas en máquinas destinadas a tal fin, utilizando *hardware* de bajo costo, como por ejemplo una Raspberry Pi [37].

En simultáneo con el desarrollo de este *framework*, el proyecto en el cual está enmarcado Mate Marote ha incorporado dos nuevos juegos: “El atrapa-números” y “Rompecabeza”. Estos juegos están siendo desarrollados por el Inserm-CEA Cognitive Neuroimaging Unit [38] y el Laboratorio de Neurociencia Integrativa y se focalizan en el entrenamiento del cálculo aritmético [25] y el planeamiento, respectivamente.

Asimismo, se prevee la inclusión de juegos para entonar capacidades cognitivas relacionadas al lenguaje y el vocabulario. Un ejemplo de este tipo de juegos es *Verbosity* [39] donde dos usuarios suman puntos cuando un usuario adivina palabras a partir de las pistas dadas por el otro. Para llevar adelante este tipo de actividades, se plantea el diseño de una infraestructura multiusuario para fomentar y estudiar la interacción entre niños. Este desarrollo deberá incluir agentes inteligentes con el objetivo de que los niños utilicen juegos de dos participantes (uno el niño, otro el agente) aun estando solos. El agente deberá ajustar su estrategia de juego teniendo en cuenta el avance y desempeño del niño como parte de una intervención educativa.

En definitiva, este proyecto abre un sinfín de oportunidades que se pueden aprovechar para conocer más sobre como piensan los niños. Con esto se pueden mejorar los recursos que se les ofrece, promoviendo de esta forma la igualdad de posibilidades. En palabras de quien fuera el presidente de la Republica Oriental del Uruguay en el 2009, Tabaré Vázquez: “El

objetivo a largo plazo del Plan Ceibal es promover la justicia social mediante la promoción de la igualdad de acceso a la información y herramientas de comunicación para todo nuestro pueblo”.

Apéndice

A. FINANCIAMIENTO

Esta tesis fue desarrollada en el marco de los siguientes proyectos e instituciones que hicieron posible su realización:

- El Departamento de Computación de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires, que facilitó el lugar de trabajo, la sala de servidores y recursos humanos que participaron.
- El proyecto *OLPC@school - Incorporating Information Technology in the Teaching and Learning Cycle* cuya financiación proviene de IBM Scalable Data Analytics, 2010 Innovation Awards.
- El proyecto *The brain's turing machine: serial chaining of cognitive operations in human and non-human primates* financiado por Human Frontiers Science Program.

Bibliografía

- [1] M. Prensky and M. Prensky. Digital game-based learning. 2008.
- [2] M.I. Posner and M.K. Rothbart. *Educating the human brain*. Amer Psychological Assn, 2007.
- [3] S. Dehaene. *The number sense: How the mind creates mathematics*. Oxford Univ Pr, 2011.
- [4] E.M. Brannon. The development of ordinal numerical knowledge in infancy. *Cognition*, 83(3):223–240, 2002.
- [5] S. Dehaene, V. Izard, E. Spelke, and P. Pica. Log or linear? distinct intuitions of the number scale in western and amazonian indigene cultures. *science*, 320(5880):1217–1220, 2008.
- [6] S. Dehaene and R. Akhavein. Attention, automaticity, and levels of representation in number processing. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21(2):314, 1995.
- [7] S. Dehaene. Précis of the number sense. *Mind & Language*, 16(1):16–36, 2001.
- [8] V. Izard, P. Pica, E.S. Spelke, and S. Dehaene. Flexible intuitions of euclidean geometry in an amazonian indigene group. *Proceedings of the National Academy of Sciences*, 108(24):9782, 2011.
- [9] Argentina. Inclusión digital educativa. <http://www.inclusiondigital.gob.ar/>.
- [10] Argentina. Conectar igualdad. <http://www.conectarigualdad.gob.ar/>.
- [11] Ciudad de Buenos Aires. Plan sarmiento ba. <http://sarmientoba.buenosaires.gob.ar/>.
- [12] One Laptop Per Child. (olpc). <http://laptop.org/>.
- [13] Salman Khan. Khan academy. <http://www.khanacademy.org/>.
- [14] Salman Khan. Khan academy. <http://khan.ceibal.edu.uy/>.
- [15] K.R. Koedinger and AT Corbett. Cognitive tutors: Technology bringing learning science to the classroom. *The Cambridge handbook of the learning sciences*, pages 61–78, 2006.
- [16] P.A. Cohen, J.A. Kulik, and C.L.C. Kulik. Educational outcomes of tutoring: A meta-analysis of findings. *American educational research journal*, 19(2):237–248, 1982.
- [17] A.C. Graesser, N.K. Person, and J.P. Magliano. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied cognitive psychology*, 9(6):495–522, 1995.

-
- [18] K.R. Koedinger, J.R. Anderson, W.H. Hadley, M.A. Mark, et al. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education (IJAIED)*, 8:30–43, 1997.
- [19] V.A. Aleven and K.R. Koedinger. An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. *Cognitive Science*, 26(2):147–179, 2002.
- [20] C.M. MacLeod. Half a century of research on the stroop effect: an integrative review. *Psychological bulletin*, 109(2):163, 1991.
- [21] D. Klahr. Solving problems with ambiguous subgoal ordering: Preschoolers’ performance. *Child Development*, pages 940–952, 1985.
- [22] A.D. Baddeley, GJ Hitch, and GH Bower. The psychology of learning and motivation. *The psychology of learning and motivation*, 8, 1974.
- [23] Django framework. Django software foundation. <https://www.djangoproject.com/>.
- [24] Laboratorio de Neurociencia Integrativa FCEN-UBA. Mate marote. <http://matemarote.com.ar/>.
- [25] A.J. Wilson, S.K. Revkin, D. Cohen, L. Cohen, and S. Dehaene. An open trial assessment of “the number race,” an adaptive computer game for remediation of dyscalculia. *Behavioral and Brain Functions*, 2(1):20, 2006.
- [26] Fedora. <http://fedoraproject.org/>.
- [27] Sugar labs. <http://www.sugarlabs.org/>.
- [28] M. Prensky. Fun, play and games: What makes games engaging. *Digital game-based learning*, pages 1–31, 2001.
- [29] Centro de Educación Médica e Investigaciones Clínicas (CEMIC). Norberto quirno. <http://www.cemic.edu.ar/>.
- [30] Sqlite. <http://www.sqlite.org/>.
- [31] CherryPy. <http://www.cherrypy.org/>.
- [32] Departamento de Computación de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires. <http://www.dc.uba.ar>.
- [33] Pyunit. <http://pyunit.sourceforge.net/>.
- [34] Google Analytics. <http://www.google.com/analytics/>.
- [35] Laboratorio de Neurociencia Integrativa. <http://neurociencia.df.uba.ar/>.
- [36] Chischicos. http://www.chischicos.com.ar/ver_notas.asp?id=187.
- [37] Raspberry pi. <http://www.raspberrypi.org/>.
- [38] INSERM-CEA Cognitive Neuroimaging Unit. <http://www.unicog.org/>.
- [39] Luis Von Ahn. Games With A Purpose (GWAP). Verbosity. <http://www.gwap.com/gwap/gamesPreview/verbosity/>.