



DESARROLLO DE UN VIDEOJUEGO EDUCATIVO PARA LA ENSEÑANZA DE PROGRAMACIÓN

ANGIE MELISSA DELGADO LEÓN CÓDIGO: 1150990

GERSON YESID LÁZARO CARRILLO CÓDIGO: 1150972

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS

SAN JOSÉ DE CÚCUTA

2014



DESARROLLO DE UN VIDEOJUEGO EDUCATIVO PARA LA ENSEÑANZA DE PROGRAMACIÓN

ANGIE MELISSA DELGADO LEÓN CÓDIGO: 1150990

GERSON YESID LÁZARO CARRILLO CÓDIGO: 1150972

PRESENTADO A:

ROSANA DÍAZ

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS

SAN JOSÉ DE CÚCUTA

2014



CONTENIDO

INTRODUCCIÓN	4
GENERALIDADES DE LA TEMATICA A TRATAR.....	5
Áreas Integradas.....	5
Importancia	6
Antecedentes	7
ANÁLISIS DEL SISTEMA.....	10
Propuesta de solución.....	10
Objetivos de la solución	12
DISEÑO DEL SISTEMA	16
Herramientas computacionales.....	16
Diagrama de clases	17
Modelo de datos.....	18
Especificación de escenarios	19
REFERENCIAS BIBLIOGRÁFICAS	36



INTRODUCCIÓN

"Todos deberían aprender como programar una computadora, porque eso te enseña a pensar" Steve Jobs

En una sociedad globalizada, cambiante en torno a las tecnologías, y en donde los computadores son una parte fundamental del mundo y de nuestra vida diaria es importante animar a la gente a dejar de ser consumidores de contenido digital y enseñarles nociones básicas sobre el funcionamiento del mundo en el que vivimos.

La computación es una inmensa área de oportunidades para el que sepa aprovecharlas. Sin embargo, aún hay cierto miedo o resistencia a involucrarse con el código cuando la verdad de las cosas, es que cada vez es más fácil comprender la programación, por lo menos a un nivel básico. Aprender a programar sirve no solo para automatizar tareas en los computadores y crear productos sino también para solucionar problemas en la vida real ya que ayuda a razonar y desarrollar un pensamiento lógico. Programar brinda la oportunidad de desarrollar ideas e implementarlas de manera creativa mediante una experiencia inmersiva y constructiva, donde cada logro impulsa a desarrollar nuevos logros más grandes.

Desde este punto de vista, surge la necesidad de implementar métodos de aprendizaje alternativos a los tradicionales, que permitan a las personas comprender los fundamentos de programación de una manera creativa, didáctica y fácil de aprender. Tomando como base esta problemática surge la idea de desarrollar Alan y el Misterioso Reino de Eniac, un juego que ilustre a través de niveles el funcionamiento de las instrucciones y principios de la programación.



GENERALIDADES DE LA TEMATICA A TRATAR

Áreas Integradas

Para la realización del videojuego “Alan y el misterioso reino de Eniac” se requiere la integración de diferentes áreas del conocimiento y de la programación, las cuales son listadas a continuación:

- Fundamentos de programación

Es la asignatura referente para el usuario, pues el videojuego está enfocado a personas que se inician en la programación, y es en esta materia en donde se encuentran las nociones básicas.

- Estructuras de datos

La utilización de diferentes estructuras de datos será fundamental para el correcto funcionamiento de la aplicación.

- Bases de datos

Los datos de los perfiles de cada usuario, además de algunos apartes del juego se apoyan en la utilización de datos procedentes de una base de datos creada para la aplicación.

- Programación web

El juego se ejecutará a través de un navegador, y hará uso del esquema cliente servidor. Por lo tanto, es necesario integrar la programación web (en este caso en php y JavaScript) con un punto adicional que será el diseño web (logrado con html5 y css3).

- Diseño gráfico

A pesar de estar fuera del pensum de ingeniería de sistemas, el videojuego utilizará una serie de gráficos en 2D para los cuales se hace necesaria la utilización de fundamentos del diseño gráfico.



Importancia

La habilidad para programar enseña a las personas a pensar analíticamente y resolver problemas (no solo computacionales, sino también matemáticos, gerenciales, administrativos, e incluso de la vida común) además de mejorar la capacidad para automatizar los procesos que tiene a cargo. Programar permite a las personas escribir nuevos tipos de cosas: cualquier aplicación de software puede convertirse en una historia interactiva, en un juego, una animación o una simulación. Por esta razón el aprendizaje de la programación es de gran trascendencia en la vida actual, más aun teniendo en cuenta que la sociedad está iniciando un nuevo proceso de alfabetización digital.

La realización de un proyecto para la enseñanza lúdica y sencilla de los conceptos básicos de la programación es necesaria como un punto de apoyo que permita formar y reforzar los conocimientos no solo de los estudiantes de Ingeniería de Sistemas sino a toda persona en general.

Alan y el Misterioso Reino de Eniac se encuentra dirigido a toda aquella persona que desee aprender a programar de una manera práctica y analítica con el objetivo de contribuir a la alfabetización digital y al desarrollo del razonamiento y el pensamiento algorítmico para motivar a los usuarios a no ser únicamente consumidores de contenido tecnológico sino formarlos para que sean capaces de desarrollar la creatividad y así comenzar a construir y a solucionar problemas, no solo en el ambiente tecnológico sino también en el mundo real.

La realización de este proyecto servirá como herramienta de apoyo tanto para estudiantes como para docentes que fortalezca y complemente el modelo presencial de aprendizaje y que ayude a desarrollar el pensamiento algorítmico para contribuir a la construcción de conocimiento y trabajos que aporten mejoras a la sociedad y que puedan generar un valor en las organizaciones. La población académica beneficiada en ingeniería de sistemas se compone de 112 alumnos distribuidos en 4 grupos con 4 docentes diferentes, los cuales corresponden al 25.5% de la población total de la carrera. Adicional a esto todos los estudiantes de otras carreras que reciben cursos introductorios a la programación. El videojuego contará con una serie de niveles que seguirán una línea de aprendizaje continuo que integrará todos los conceptos que se vayan aprendiendo en el transcurso de la historia y que les permitirá hacer un retroceso para reforzar conocimientos en los cuales se hayan notado falencias.



Antecedentes

El aprendizaje de la programación de manera didáctica no es una idea nueva en ambientes educativos. El gran auge de las tecnologías ocurrido en los últimos años, ha conllevado una creciente demanda de personal capacitado en programación y áreas afines. Sin embargo, una barrera importante sigue siendo el primer acercamiento a un lenguaje de programación: Suele verse como algo complicado, exhaustivo e incluso ilógico. Para superar estas barreras, varios proyectos independientes (académicos y no académicos) han apostado por nuevas maneras de enseñanza, que permitan ver como una actividad atractiva el desarrollo de un programa de software.

En primer lugar, la universidad Carnegie Mellon en Pensilvania, Estados Unidos ha creado un software lúdico y educativo llamado Alice, el cual mediante un completo entorno 3D y su propio lenguaje del mismo nombre intenta enseñar a los estudiantes los fundamentos de programación de una manera visual y accesible a cualquier tipo de usuario. En su intención de hacer más fácil el aprendizaje de la programación, dicha universidad ha liberado esta herramienta como software libre, para que pueda ser implementada por profesores y estudiantes en cualquier lugar del mundo. Estudios realizados en el Ithaca College y en Saint Joseph's University revelan que las notas medias de estudiantes sin experiencia de programación y en su primer curso de informática subieron de suficiente a notable y que su retención pasó del 47% al 88%.

La universidad nacional de Salta, Argentina, ha propuesto un método propio de programación lúdica. Para ello, utiliza justamente el software Alice mencionado anteriormente combinado con Diagramar, un software para crear y ejecutar diagramas de bloques. El software junto a las asesorías brindadas por los maestros y los pares educativos, fundamentan su proyecto llamado “La programación lúdica como estrategia de articulación entre niveles”.

Otra propuesta interesante es Scratch. Scratch es un proyecto del Grupo Lifelong Kindergarten del Laboratorio de Medios del MIT que ayuda a los jóvenes a aprender a pensar creativamente, razonar sistemáticamente, y trabajar colaborativamente – habilidades esenciales para la vida en el siglo XXI – brindándoles la oportunidad de programar historias interactivas, juegos y animaciones que se pueden compartir con otros en la comunidad en línea. Gracias al apoyo de la Motorola Foundation, Motorola de Colombia Ltda., Motorola Solutions Foundation y la gestión de la ONG Give to Colombia, la Fundación Gabriel Piedrahita Uribe (FGPU) implementó el proyecto Scratch con el que buscó contribuir al desarrollo de habilidades del siglo XXI, capacidades intelectuales de orden superior y pensamiento computacional, por parte de estudiantes de básica primaria en Colombia.



MIT ha mezclado la didáctica con las aplicaciones reales, creando App Inventor. Esta aplicación permite crear apps reales para Android solo arrastrando y soltando bloques de código.

En Colombia también es importante destacar el proyecto “CUPI2”: Una solución integral al problema de enseñar y aprender a programar desarrollado por la Universidad de los Andes de Colombia que a través de cuatro estrategia – aprendizaje activo, aprendizaje basado en problemas, aprendizaje incremental y aprendizaje basado en ejemplos – busca generar una gran cantidad de habilidades asociadas con cualquier profesional en ingeniería (aprender a entender un problema, a plantear soluciones efectivas, a manejar lenguajes para expresar una solución, a utilizar herramientas que entiendan esos lenguajes, a probar que la solución sea válida, a justificar las decisiones tomadas, etc.).

Otra iniciativa en Colombia es la corporación Ruta N creada por la Alcaldía de Medellín, UNE y EPM que facilita la evolución económica de la ciudad hacia negocios intensivos en ciencia, tecnología e innovación, de forma incluyente y sostenible. Los lineamientos bajo los cuales trabajan son: cultura de la innovación, gestión del conocimiento, gestión de redes, acceso a mercados, acceso a capital e innovación empresarial.

A nivel menos académico, la empresa privada y fundaciones sin ánimo de lucro se han involucrado también en la enseñanza de la programación. De esta manera han surgido muchas iniciativas, de las cuales mencionamos las más relevantes en la actualidad: Codecademy, con un editor web y proyectos para desarrollar paso a paso en diferentes lenguajes de programación, de manera interactiva, fácil y altamente entendible se ha convertido en uno de los proyectos web de programación con más visitantes en los últimos meses.

CheckiO es también una plataforma bastante interesante de analizar. CheckiO nace en Ucrania como una StartUp tecnológica con base en Las Vegas y su objetivo principal es ayudar a las personas a desarrollar competencias en Python con el apoyo de una gran comunidad a través de un videojuego por niveles. Los jugadores pueden recibir ayuda de las demás personas y recibir revisiones de código de sus soluciones. Similares a las clases de programación, existe un ciclo de retroalimentación con expertos de código en Python que guiarán a los estudiantes para que continúen aprendiendo.

Los grandes magnates de la industria del software, han apoyado la enseñanza interactiva de la programación uniéndose para crear su propio proyecto conjunto: Code.org. Entre sus promotores se encuentran Mark Zuckerberg (creador de Facebook), Bill Gates (fundador de Microsoft), Jack Dorsey (creador de Twitter), Ruchi Sanghvi (vicepresidenta de Dropbox) entre muchos otros, quienes además hacen la labor de profesores en algunos videos de la plataforma.

FightCode por otro lado es una plataforma que ayuda a mejorar las habilidades en Javascript, pero programando robots. Cada robot podrá hacer uso de una API, que le



permitirá buscar oponentes, disparar, sortear obstáculos y mucho más. El juego no enseña a programar en Javascript, por lo que es necesario tener algunos conocimientos antes de empezar a configurar un robot. Una vez se tiene el robot listo, se pone a prueba luchando contra los robots de otros usuarios.

La idea de acoplar los videojuegos como método de enseñanza ha sido utilizada también por Tynker (<http://www.tynker.com>). Este proyecto web, enseña los conceptos de la programación por medio de diversos videojuegos en los que la interactividad y la inmersión en la trama de la historia juegan un papel fundamental. Otros proyectos por el mismo estilo se pueden encontrar en CodeCombat (www.codecombat.com/) y Light bot (www.light-bot.com/hocflash.html).

Estas iniciativas han cobrado tanta relevancia en el internet, que las iniciativas anteriormente mencionadas (Code.org, Tynker, codecombat, light bot, entre otras) se han unido en una estrategia llamada “La hora del código”, consistente en un reto para los usuarios de dichas aplicaciones: dedicar una hora de cada día a aprender código. No se hace énfasis en un lenguaje específico, sino en la programación como tal, haciendo al usuario parte de la plataforma y logrando con él una interacción diaria. Al completar la hora de código, puede incluso recibir una certificación digital acreditando su hora de código, a medida que resuelve problemas didácticos cada vez más complejos.

Tras el éxito de code.org han surgido nuevas iniciativas con la misma intención. Ejemplo de ella es ThoughtSTEM, una startup surgida con el fin de crear videojuegos al estilo minecraft con el mismo sistema de arrastrar y soltar.

Play-i por su parte enseña a programar robots arrastrando y soltando bloques. El mecanismo es el mismo que code.org, pero en vez de programar juegos en pantalla, programar robots físicos que pueden moverse y realizar diferentes funciones, mezclando la programación con la robótica.

Apoyando estas iniciativas, google ha creado su propia librería de código libre, con bloques personalizables para utilizar en videojuegos de este tipo. No es un juego como los anteriores, al contrario, es un conjunto de bloques que los programadores pueden utilizar en sus propios videojuegos.

De esta manera, diversos actores externos han desarrollado temáticas que contribuyen y refuerzan la idea de enseñar a programar de manera lúdica. Estas iniciativas sirven de base a Alan y el Misterioso Reino de Eniac para la consecución de sus objetivos.



ANALISIS DEL SISTEMA

Propuesta de solución

Alan y el Misterioso Reino de Eniac será un videojuego desarrollado en un ambiente web para el aprendizaje didáctico de los conceptos básicos de la programación. Mediante una trama ficticia inspirada en la edad media y adaptada rigurosamente a la programación, el estudiante podrá entender los principales componentes del código y deberá utilizarlos correctamente para poder avanzar a los siguientes niveles, cada uno de los cuales contará con nuevas temáticas y una mayor dificultad que integre los conocimientos ya adquiridos.

Como ya ha sido mencionado, la organización del juego se lleva a cabo mediante niveles, que inducen poco a poco temas más complejos, y aumentan la dificultad conforme se superan los diferentes obstáculos encontrados. Cada nivel en el juego presenta un tema central de aprendizaje: Variables, métodos, condicionales, ciclos, entre otros. Sin embargo, en cada nivel podrán -y en muchos casos deberán- utilizarse los temas de los niveles anteriores para lograr su consecución y por tanto, el avance al siguiente nivel. Además, cada uno de estos estará dividido en subniveles o etapas en los cuales la dificultad aumentará gradualmente, bajo la misma temática. En el camino el jugador recolecta puntos, con lo que podrá conocer su evolución y sus logros y además, compararse con respecto a los demás usuarios en el ranking del juego. Pero no será esta la única manera de medir el avance: también se otorgan medallas virtuales a los jugadores que superen determinadas pruebas, o logren un objetivo específico planteado en alguna etapa del juego.

Cada usuario poseerá su propio perfil, con algunos datos relevantes en la estructura del juego: su nivel y etapa actual y puntaje obtenido hasta el momento. La información sobre la ubicación actual de cada usuario en el juego quedará guardada para las próximas sesiones, con solo introducir su usuario y contraseña. De esta manera, los avances no se pierden al cerrar la página o cambiar de equipo, sino que se mantienen en la plataforma para continuar justo en donde se dejó la última vez. El videojuego como tal estará desarrollado en las modernas tecnologías web funcionando directamente en el cliente (html5 para la maquetación, css para el diseño y JavaScript para la lógica del programa, específicamente mediante la utilización del elemento canvas), mientras las operaciones de registro, inicio de sesión, y generación de perfiles se realizarán mediante PHP, en el servidor.

En primera medida, el juego está dirigido a la población de semestres iniciales tanto de ingeniería de sistemas, como de aquellos programas que incluyen cursos básicos de programación en su pensum. De esta manera se llevará a cabo un valioso aporte académico a los estudiantes que se inician en la programación (especialmente



aquellos que se encuentren cursando la asignatura “fundamentos de programación”) funcionando como un complemento lúdico y práctico al aprendizaje recibido en el aula de clases, perfeccionando sus conocimientos en los temas tratados y comprendiéndolos de manera visual e inmersiva. En etapas posteriores, el videojuego podrá generalizarse en su alcance no sólo a estudiantes, sino a todas aquellas personas que deseen aprender a programar desde cero, o servir de repaso a quienes ya tienen un conocimiento básico en el tema.

Para que la aplicación logre una aceptación entre la comunidad educativa, se han planteado reuniones con docentes que dictan fundamentos de programación para que ellos aporten sus propias ideas y opiniones con respecto a la plataforma, y en base a esto establecer los requerimientos que se deben cumplir. Concretamente han sido entrevistados dos docentes. A continuación se resumen sus aportes:

- Ing. Fredy Vera: Al iniciar en fundamentos de programación, la principal falla que tienen los alumnos está en la lógica. La programación representa en la mayoría de estudiantes un campo aún inexplorado al cual es difícil enfrentarse, especialmente por la abstracción necesaria para llegar a la resolución de un problema, y cualquier ayuda lúdica que permite una introducción secuencial al tema podría mejorar sustancialmente la manera en la que se asimilan estos conceptos. El pensamiento algorítmico de los estudiantes puede ser capacitado de maneras interactivas que aún no están siendo implementadas en la Universidad. Además, el juego no debería estar solo planteado como una estrategia extracurricular, sino integrarse también en las clases. La tercera nota de los estudiantes podría contener un ítem por alcanzar determinado punto en el juego en una fecha establecida. Para esto, será de gran ayuda que los docentes puedan también monitorear los progresos de sus estudiantes en el juego.
- Ing. Boris Pérez: La aplicación puede ser una buena herramienta de ayuda en el proceso de aprendizaje. Sin embargo, el docente prefiere establecer una nueva reunión en un momento más avanzado del proyecto, cuando se cuente con un prototipo funcional que pueda ser evaluado en cuanto a su potencial, y solo en ese momento establecer aspectos a mejorar.

Gracias a estas consultas se han podido validar los requerimientos de la solución además de agregar uno nuevo que permita el control de los docentes sobre los estudiantes de sus cursos.



Objetivos de la solución

NOMBRE	RF-1 REGISTRO USUARIO
RESUMEN	El sistema debe permitir el registro y creación de perfiles de usuario.
ENTRADAS	Nombre, correo, nombre de usuario, contraseña, universidad, carrera, semestre, entre otros que puedan resultar relevantes.
RESULTADO	Usuario registrado en la base de datos y un perfil visual en la aplicación.

NOMBRE	RF-2 GUARDAR PROGRESO USUARIO
RESUMEN	El sistema debe almacenar el progreso de los usuarios así como las medallas que se le otorguen
ENTRADAS	Nivel Actual
RESULTADO	Acceso a nuevos niveles, actualización de los datos del usuario así como de su perfil.

NOMBRE	RF-3 AUTENTICAR USUARIO
RESUMEN	El sistema debe permitir la autenticación de los jugadores que acceden al portal y, dependiendo del jugador actualizar el mapa de niveles.
ENTRADAS	Nombre de Usuario, contraseña.
RESULTADO	Si los datos ingresados son correctos permitirá el acceso a la plataforma.

NOMBRE	RF-4 RECUPERAR CONTRASEÑA
RESUMEN	El sistema debe contener mecanismos para que el usuario pueda



	generar una nueva contraseña en caso de perder la anterior.
ENTRADAS	Correo electrónico.
RESULTADO	Un mensaje al correo electrónico con un link que permita la creación de una nueva contraseña.

NOMBRE	RF-5 GENERAR MAPA
RESUMEN	El sistema debe generar un mapa de los niveles que incluya acceso a los niveles permitidos y en el cual el jugador se ubique de manera sencilla. La plataforma deberá actualizar dinámica y automáticamente este mapa de acuerdo al progreso de los jugadores.
ENTRADAS	Nombre de usuario, nivel.
RESULTADO	Un mapa actualizado para cada jugador en la página principal del juego.

NOMBRE	RF-6 VALIDAR ACCESO A NIVEL
RESUMEN	El sistema debe verificar e informar a los usuarios a qué niveles puede acceder, y si necesitan requisitos previos para acceder a otros niveles, los debe guiar de manera automática en este proceso.
ENTRADAS	Nombre de usuario, nivel.
RESULTADO	Si es permitido el acceso se cargará este nivel, de lo contrario se notificará los requisitos que se deben cumplir para acceder.

NOMBRE	RF-7 PUBLICAR RANKING
RESUMEN	El sistema debe permitir la publicación de un ranking con los nombres de los usuarios que tengan el mayor progreso dentro del juego.
ENTRADAS	Datos de los usuarios (obtenidos desde la base de datos)
RESULTADO	Mostrar el ranking en la página principal del juego.



NOMBRE	RF-8 AVANCE POR NIVELES
RESUMEN	El sistema debe contener un sistema de niveles que facilite la comprensión de los conceptos a explicar dentro del juego.
ENTRADAS	Ubicación actual del usuario en el juego
RESULTADO	Generación del mapa propio del nivel, y avance al nivel superior una vez alcanzados los objetivos del actual.

NOMBRE	RF-9 GENERAR AYUDAS
RESUMEN	El sistema debe contar con un mecanismo de ayudas didácticas para facilitar el progreso del jugador.
ENTRADAS	Elemento sobre el cual se requiere la ayuda
RESULTADO	Mostrar información que permita el progreso del jugador.

NOMBRE	RF-10 VALIDAR ALGORITMOS
RESUMEN	El sistema debe validar los algoritmos ingresados por los usuarios en cada nivel para permitir el progreso dentro del juego.
ENTRADAS	Algoritmo propuesto por el usuario
RESULTADO	Si el algoritmo es válido se permitirá el acceso a un nuevo nivel. Se notificará al usuario.

NOMBRE	RF-11 INTEGRAR REDES SOCIALES
RESUMEN	El sistema debe incorporar funcionalidades nativas de aplicaciones de redes sociales (Facebook, Twitter, Google+), a través de vínculos como: "Comparte tu progreso con los demás", "recomiéndanos", "me gusta".
ENTRADAS	No Aplica
RESULTADO	Publicación en las redes

NOMBRE	RF-12 VISUALIZAR PROGRESO ESTUDIANTES
---------------	---------------------------------------



RESUMEN	El sistema debe permitir que los docentes de programación observen el progreso de sus estudiantes en el juego para apoyar el desarrollo
ENTRADAS	Datos de los usuarios registrados en un curso
RESULTADO	Tabulación con los resultados de los estudiantes



DISEÑO DEL SISTEMA

Herramientas computacionales

El espacio de trabajo para el desarrollo del videojuego conlleva la utilización de diferentes herramientas. Cabe destacar que el juego es desarrollado para ambientes web, realizando en el servidor todos los aspectos relacionados al sistema de usuarios (inicio de sesión, visualización de perfiles, guardado de puntajes, ranking y visualización de medallas), y en el cliente el videojuego propiamente dicho. Así pues, se mencionan a continuación las herramientas a utilizarse:

Lenguaje de programación: Se utilizarán dos lenguajes de programación: PHP para llevar a cabo las labores del servidor, y javascript para programar el juego en el cliente web. La interfaz de usuario estará complementada por HTML y CSS (lenguajes de hipertexto y cascada de estilos) para complementar la experiencia de usuario de manera atractiva.

Herramientas de Desarrollo: En lugar de un IDE, el trabajo será desarrollado a través de dos editores de texto con características adaptables al desarrollo web: Sublime Text 2, y Brackets.

Software de edición gráfica: Para los personajes y escenarios, es necesario utilizar software de retoque de imágenes y edición vectorial, en este caso Gimp e Inkscape.

Componentes especiales de programación: La aplicación será trabajada sobre el servidor XAMP (Independiente del sistema operativo, Apache como servidor de aplicaciones web, MySQL como sistema de gestión de base de datos y PHP en el desarrollo), tanto durante su desarrollo de manera local, como en su implementación sobre un servidor real. Además de esto, la comunicación entre el juego en el cliente y los datos en el servidor se realiza mediante objetos JSON (JavaScript Object Notation) y la comunicación asíncrona para el envío y recepción de JSON se lleva a cabo utilizando AJAX (Asynchronous JavaScript And XML).



```

classDiagram
    class Usuario {
        +codigo:int
        +nombre:String
        +privilegio:byte
        +descripcion:String
        +contraseña:String
        +puntaje:int
        +privilegio:int
        +usuario():void
        +obtenerDatos():String
        +actualizarDatos(codigo, nombre, descripcion):void
        +cambiarContraseña(contraseña):void
        +agregarMedalla(Medalla):void
        +registrarCurso():void
    }
    class Juego {
        +mapa:Mapa
        +juego():void
        +iniciar():void
    }
    class Medalla {
        +logro:String
        +tipo:String
        +Medalla():void
        +mostrar():String
    }
    class Mapa {
        +niveles[]:Nivel
        +instruccion:String
        +mapa():void
        +cargarGraficos():void
        +mostrar():void
    }
    class Nivel {
        +escenario:Escenario
        +dificultad:int
        +tema:String
        +nivel():void
        +mostrarNivel():void
        +avanzar():void
        +ubicarPersonaje():void
    }
    class Escenario {
        +nombre:String
        +espacio:media
        +escenario():void
        +cargarGraficos():void
    }
    class Tema {
        +nombre:String
        +explicacion:Explicacion
        +tema():void
        +asignarExplicacion():void
    }
    class Personaje {
        +sprite:media
        +personaje():void
        +cargarGraficos():void
    }
    class Curso {
        +nombre:String
        +codigo:int
        +curso():void
        +agregarAlumno():void
        +generarReporte():String
    }
    class Explicacion {
        +descripcion:String
        +ayuda:String
        +dificultad:int
        +explicacion():void
    }
    class Secundario {
        +tipo:String
        +secundario():void
    }
    class Principal {
        +nombre:String
        +principal():void
    }

    Usuario "*" -- "1" Juego
    Usuario "*" -- "*" Medalla
    Usuario "*" -- "1" Mapa
    Usuario "*" -- "1" Nivel
    Usuario "*" -- "*" Curso
    Juego "1" -- "1" Mapa
    Mapa "1" -- "1" Nivel
    Nivel "1" -- "1" Escenario
    Nivel "1" -- "1" Tema
    Nivel "1" -- "1" Personaje
    Tema "1" -- "1" Explicacion
    Personaje "1" -- "1" Secundario
    Personaje "1" -- "1" Principal
    Curso "1" -- "1" Secundario
    Curso "1" -- "1" Principal
    Explicacion "1" -- "1" Secundario
    Explicacion "1" -- "1" Principal
  
```

Figura 1. Diagrama de clases Alan y el Misterioso Reino de Eniac.



Modelo de datos

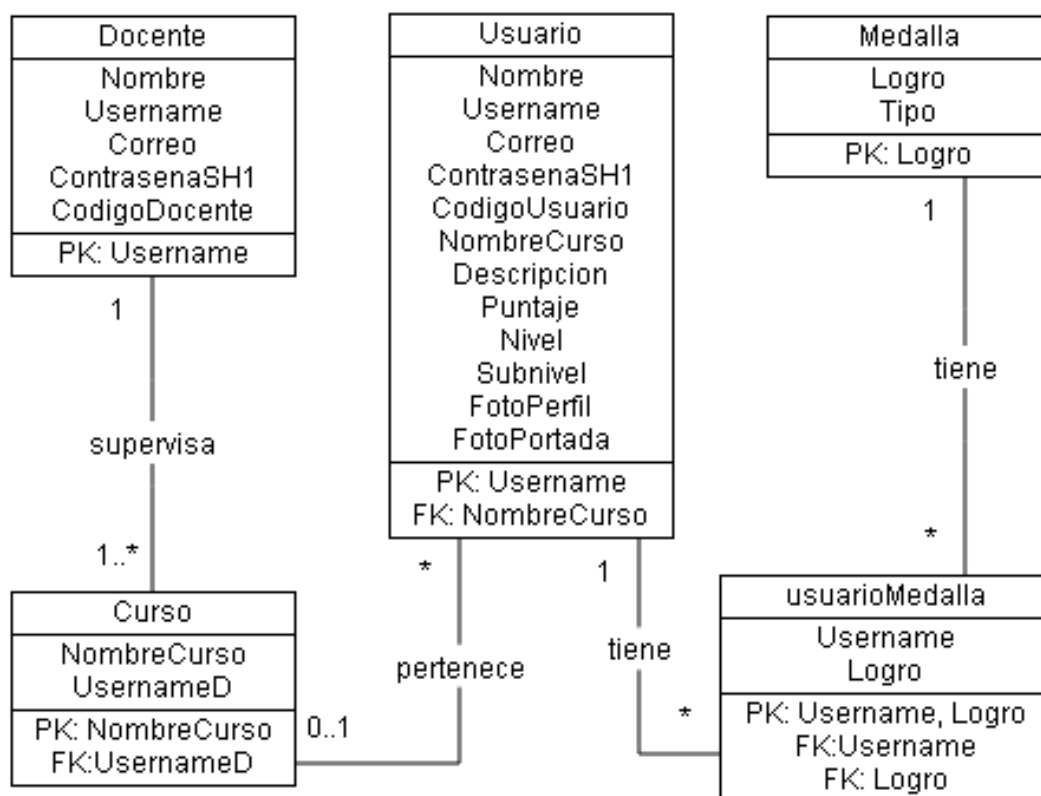


Figura 2. Modelo Lógico de Alan y el Misterioso Reino de Eniac.



Especificación de escenarios

Esquema

El diseño de la página se ha planteado de manera que se ajuste a distintos tamaños de pantalla sin afectar su correcta visualización, además de dejar el mayor espacio de disponible para el área de trabajo, que es el lugar que debe acaparar la atención del usuario durante el juego. La interfaz de usuario se realiza en diseño “flat” o plano, buscando ser elegante y visualmente atractivo, pero sin distraer al usuario en detalles cargados, dejando así su atención fija en el juego.

Encabezado (Header): El encabezado busca tener un diseño sin distracciones. Mantiene un color plano, el logo en la parte central y una barra de búsqueda semitransparente en el sector derecho desde el cual pueda buscarse a otros usuarios.

La zona izquierda de este menú se integra con la barra lateral (menú de navegación) de tal manera que al estar activa sea visible la imagen y nombre de usuario, y al estar oculta muestre solo la imagen.



Figura 3. Encabezado con el logo (provisional) en el centro, la imagen del usuario a la izquierda, y la búsqueda a la derecha.

Zona de Menús: La barra lateral izquierda cumple la función de menús. Esta se ha diseñado de tal manera que sea visible y entendible todo el tiempo, ocupando el espacio mínimo posible, lo cual se logra colocando iconos en vez de texto (cada icono guarda relación con la acción que realiza). Sin embargo, por si esto no es suficiente, cuando el usuario coloca el puntero del mouse o dispositivo apuntador sobre esta barra, se despliega junto a cada icono un botón más amplio con la definición exacta de su función. Si el usuario hace clic sobre esta barra (no sobre un botón) esta queda activa aun después de mover el puntero a otro lado. En caso contrario, la barra se repliega a su tamaño original inmediatamente después de quitar el mouse.

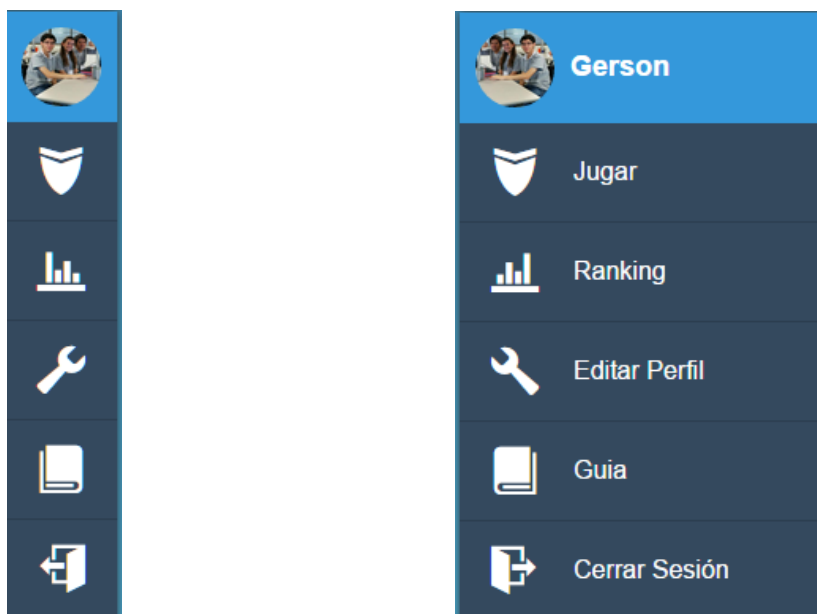


Figura 4. a) Zona de Menús oculto.

b) Zona de Menús visible

Zona de trabajo: Ocupa casi toda el área de la aplicación. Esto es intencional, dado que aquí se cargará el juego y éste necesita de un espacio considerable. Está justo debajo del encabezado, y a la derecha de la barra de menús. Ocupa casi todo el ancho de la pantalla: entre 80% y 93.5% de la pantalla (varía dependiendo de la barra de menús que puede estar desplegada u oculta. En él se cargará el juego, pero también dependiendo de la sección en la cual se encuentre el usuario, podrán cargarse: Un perfil (propio o de otro usuario), el ranking, el modo “editar perfil” o las guías.

En el modo juego, la zona de trabajo contendrá en primer lugar el mapa con los respectivos niveles, y al entrar a cada uno de ellos el espacio de juego donde el usuario podrá interactuar con la aplicación mientras lleva a cabo su proceso de aprendizaje. La interacción se realiza mediante contacto directo, donde el usuario por medio de procesos de arrastrar y soltar será instruido en la algoritmia y sus principales componentes.

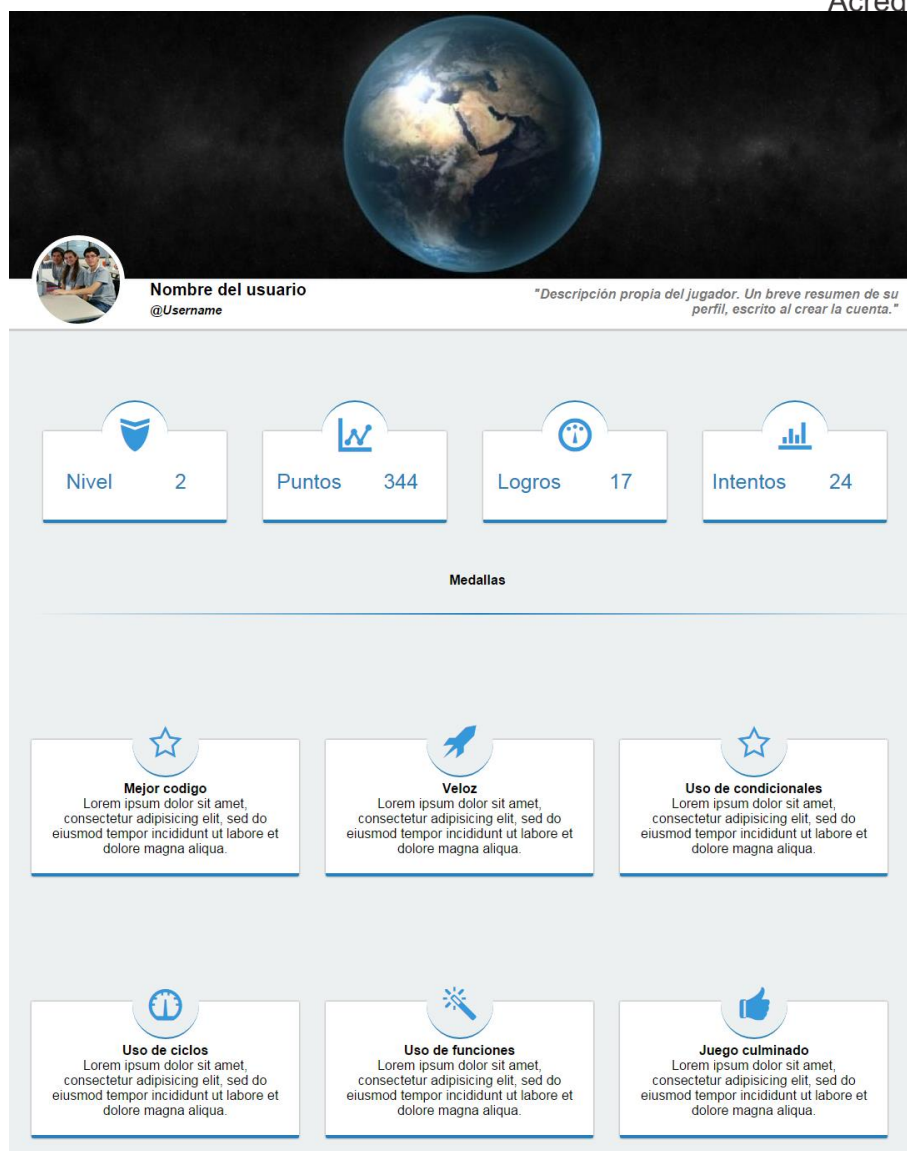


Figura 5. Área de trabajo cargando el perfil de usuario.

Zona de ayuda: Esta se divide en dos partes: En cuanto a la plataforma, y en cuanto al juego. Para la plataforma, la zona de menús cuenta con un botón llamado “guías”, el cual llevará al usuario a una zona de ayuda con temas referidos al curso (aquellos por los que ya haya pasado en el juego). Y en cuanto al juego, la zona de ayuda está inmersa en la aplicación. Al ser un juego interactivo, este paso a paso irá mostrando al usuario la ayuda necesaria para continuar.

Pie de página (footer): Al final de la página, ocupando el ancho del contenedor se encuentran los datos de fecha, universidad y creación.



Figura 6. Area de pie de página.

Colores a emplear

Para lograr armonía entre los colores, el diseño se realiza siguiendo la regla cromática “monocromática”, es decir, distintas intensidades del mismo color. El color base ha sido una tonalidad especial de azul, y sus variaciones para el diseño van desde una tonalidad casi blanca, hasta una tonalidad gris fuerte. Los colores utilizados se presentan en la siguiente paleta:

RGB	(236,240,241)	(52,152,219)	(41,128,185)	(52,73,94)	(139,160,204)
HEX	ECF0F1	3498D8	2980B9	34495E	8BA0CC

Figura 7. Tabla de colores a emplear

Tipos de letras

La interfaz de usuario en la plataforma está totalmente tipografiada con la fuente “Helvética”. Esta fuente se caracteriza por ser Sans-serif, por lo tanto no tiene remates en sus esquinas, lo cual trae como ventajas una mejor visualización en pantalla y una menor pixelación. La desventaja de usar una fuente Sans-Serif en comparación con las Serif es que pueden agotar la vista en textos extensos. Sin embargo, en este caso no hay presencia de ningún texto extenso por lo cual no conlleva problema.

Ya en la interfaz interna del juego, es necesaria la utilización de una fuente Serif, pues estas tienen un aspecto más antiguo, lo cual encaja en la trama medieval. La fuente elegida para estos efectos es la “Book Antiqua”.

Utilizando diferentes variaciones en el estilo y tamaño de estas dos tipografías, no se considera necesario la inclusión de más fuentes.



Estandarización de Botones

Desde la página inicial, el usuario podrá acceder a dos botones: Registrarse (crea un usuario en el sistema) o iniciar sesión (abre la cuenta de un usuario existente). Una vez en la plataforma, el usuario cuenta en la barra lateral con los siguientes botones:

- Usuario: Este botón carga con el nombre de usuario y su foto, y lleva a la persona directamente a su perfil. Por su relevancia es el único que tiene un color diferente a los demás del menú.

Los siguientes botones tienen un diseño similar: Color azul-gris oscuro, un icono representativo y su nombre, en un cuadrado plano y sin bordes.

- Jugar: Lleva al usuario directamente al juego
- Ranking: Muestra los mejores jugadores
- Editar perfil: Permite al usuario modificar sus datos
- Guía: Muestra al usuario las guías de los niveles avanzados en el juego
- Cerrar sesión: Vuelve a la página inicial, cerrando la sesión activa.

Al interior del juego se presentan botones para avanzar de niveles, validar el código, y volver atrás. Estos tienen un diseño medieval para ser coherentes con el juego representado. Las demás botones están inherentes en el juego: El mismo mapa los contiene, no como botones fijos, sino como imágenes en el contexto de la aplicación que crean una mayor experiencia de usuario.



Prototipo de las interfaces principales



Figura 8. Página de Inicio



Figura 9. Página de Juego

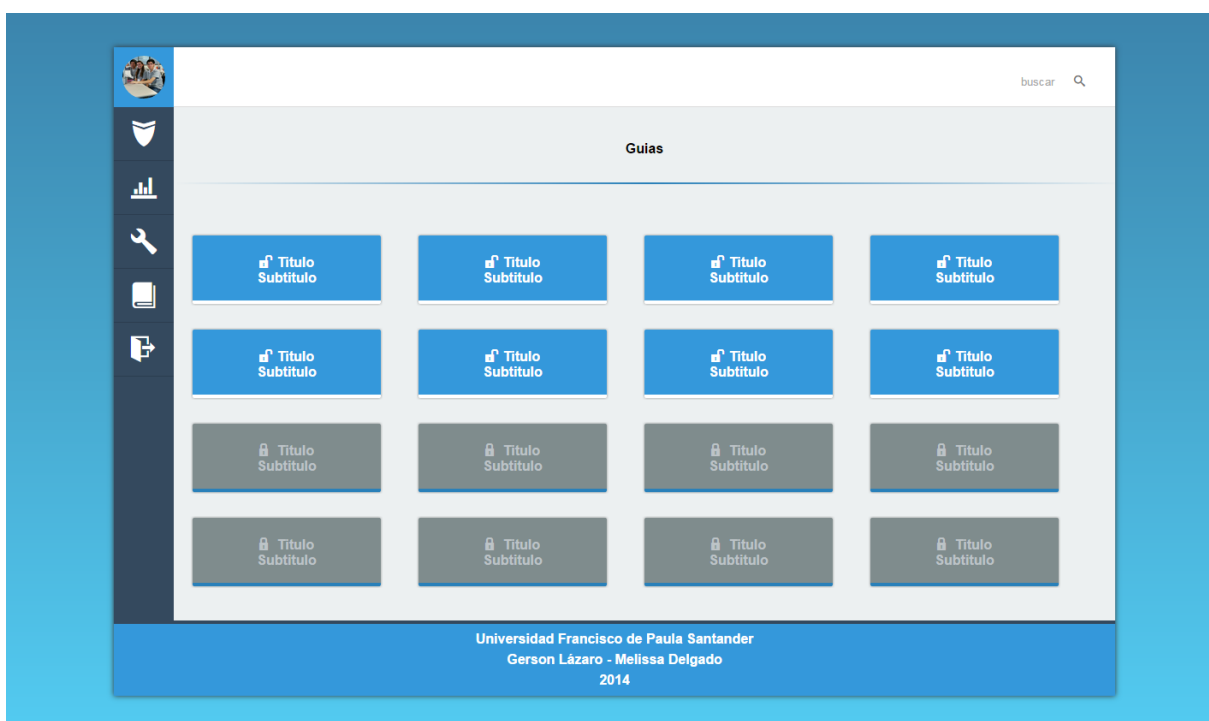


Figura 10. Guía de Niveles

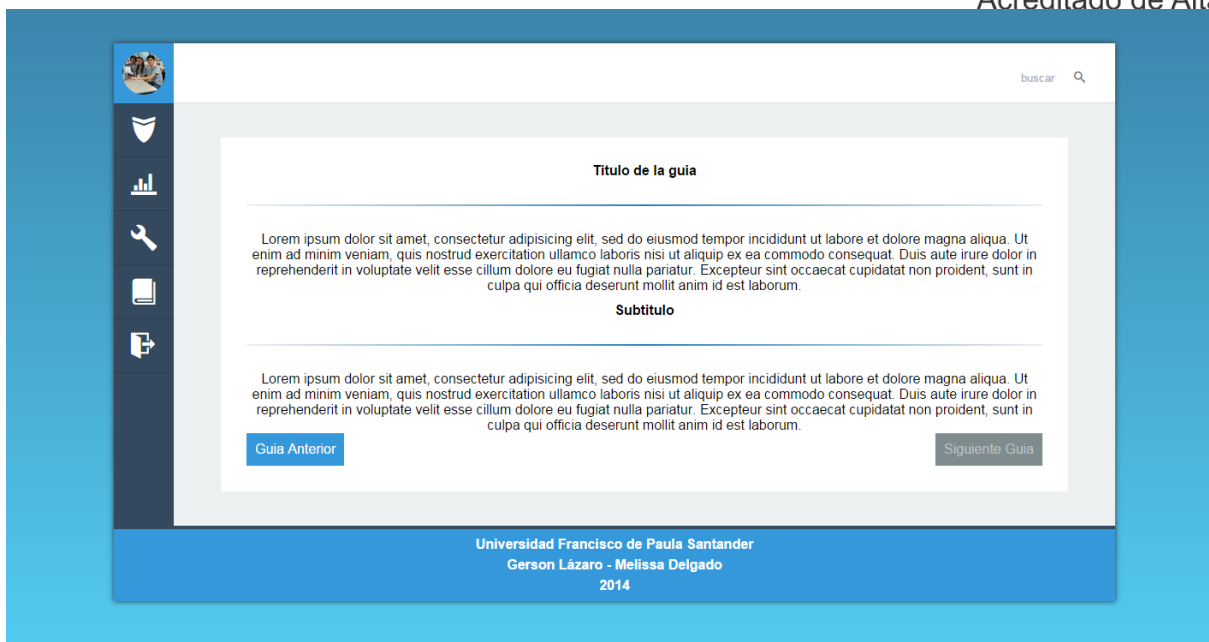


Figura 11. Guía de Explicación Nivel

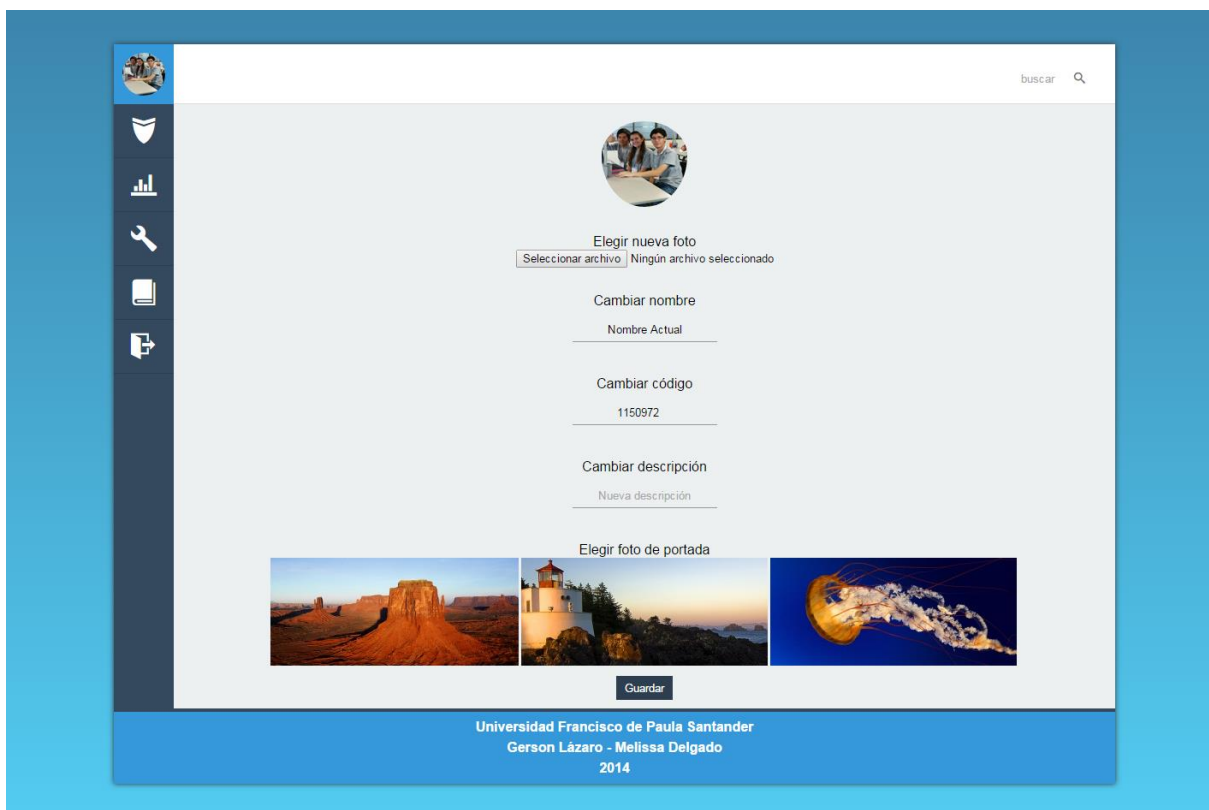


Figura 12. Editar Perfil

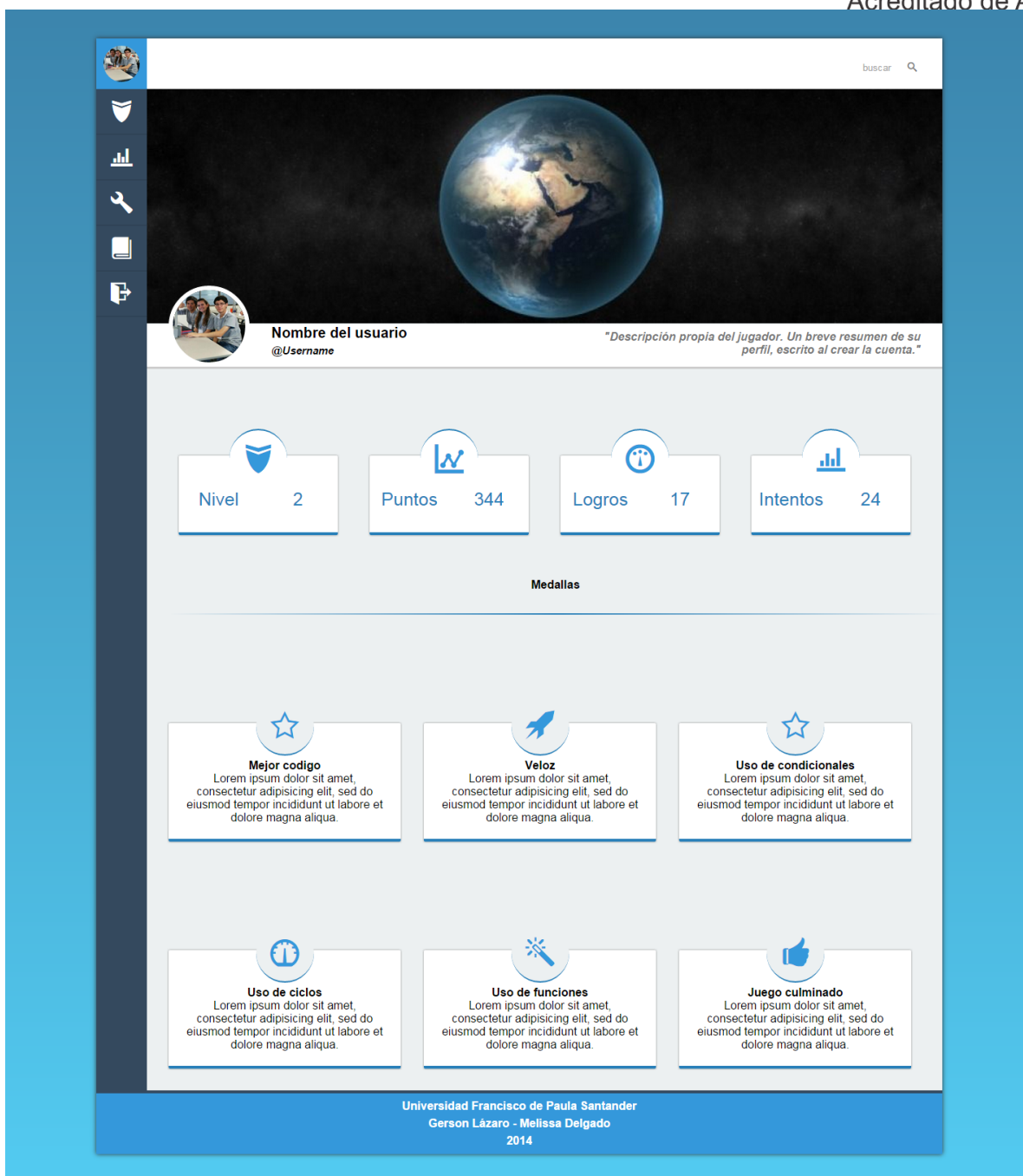


Figura 13. Perfil de Usuario



Ranking			
#	Usuario	Nivel	Puntaje
1	Usuario 1	9	1098
2	Usuario 2	8	1078
3	Usuario 3	8	1002
4	Usuario 4	8	931
5	Usuario 5	7	910
6	Usuario 6	5	807
7	Usuario 7	5	800
8	Usuario 8	5	752
9	Usuario 9	3	751
10	Usuario 10	2	721
11	Usuario 11	2	714
12	Usuario 12	1	201
13	Usuario 13	1	142
14	Usuario 1	9	1098
15	Usuario 2	8	1078
16	Usuario 3	8	1002
17	Usuario 4	8	931
18	Usuario 5	7	910
19	Usuario 6	5	807
20	Usuario 7	5	800
21	Usuario 8	5	752
22	Usuario 9	3	751
23	Usuario 10	2	721
24	Usuario 11	2	714
25	Usuario 12	1	201
26	Usuario 13	1	142

Universidad Francisco de Paula Santander
Gerson Lázaro - Melissa Delgado
2014

Figura 14. Ranking



Solución algorítmica

Alan y el reino de Eniac sigue el patrón de arquitectura de software MVC (Modelo-Vista-Controlador) implementado de la siguiente manera:

- **Modelo:** Las clases del modelo son las únicas que tienen acceso a la base de datos. Existe una clase modelo que contiene todos los métodos necesarios para conectar, desconectar, escribir, modificar y borrar en la base de datos, y las demás clases del modelo heredan de ella, para realizar operaciones específicas de cada clase.
- **Vista:** La vista está compuesta de pequeños módulos de código que componen la interfaz gráfica de usuario.
- **Controlador:** Realiza todas las operaciones de la plataforma.
- **Enrutador (index.php):** Toma las peticiones del usuario y llama a la función del controlador encargada de ejecutarla.
- **Estáticos:** Archivos Css, imágenes y JavaScript que se usaran en la aplicación pero que por no estar generados dinámicamente, simplemente se han ubicado en una carpeta especial y pueden usarse cuando sean necesitados por alguna vista. Se ubican estratégicamente junto al index.php para su fácil acceso.

Cuando el usuario realiza una petición esta es recibida por el index.php (todas las interacciones del usuario se realizan directamente con el index.php, las demás url's que pueden aparecer en el juego son generadas dinámicamente). El index.php define que acción solicitó el usuario (utilizando las variables de sesión, get y post) y llama al controlador indicado a realizar dicha acción. El controlador se comunica con el modelo para obtener los datos que sean necesarios (o modificarlos) y si debe dar una respuesta la guarda en variables temporales. Posteriormente, el controlador arma la interfaz a mostrar, tomando los fragmentos necesarios desde las vistas, e integrándolos con las variables locales que debe mostrar. Esta vista completa es procesada y compilada con php para mostrarse finalmente en el mismo index.php.

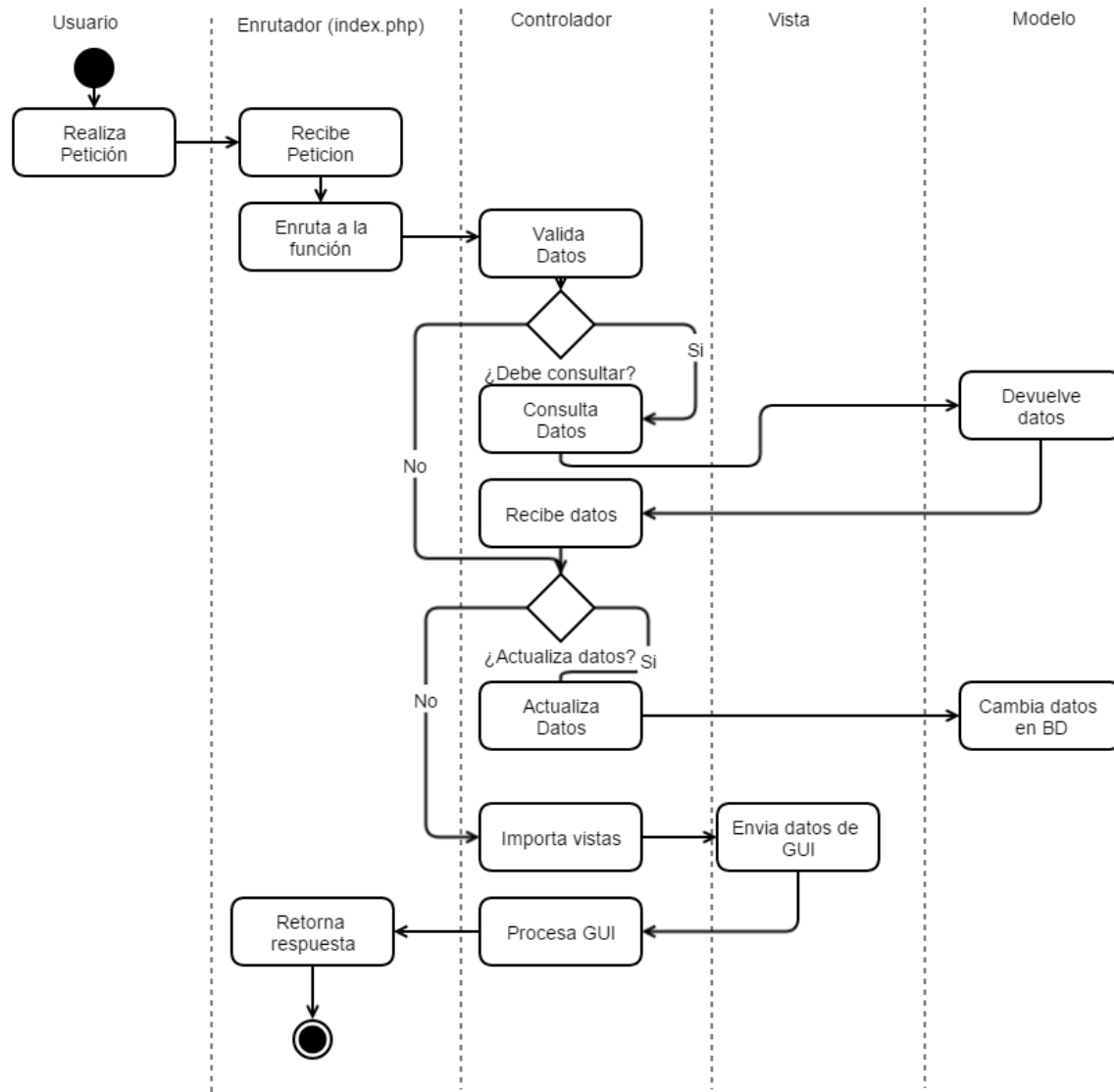


Figura 15. Diagrama de actividades general

Control del Usuarios

Para el registro de un usuario, este puede o no ser estudiante. En el primer caso, debe indicar su código, y si el docente ha creado un grupo puede seleccionarlo desde una lista desplegable. De lo contrario solo indica sus datos personales.

Para ingresar a la plataforma es requerida una cuenta y contraseña, que se crea durante el registro. Los datos del usuario son guardados en una base de datos MySQL por la aplicación como cadenas de texto o números según corresponda. El único dato que recibe un trato especial es la contraseña, la cual se encripta mediante



SHA1 antes de guardarse en la base. Al iniciar sesión, los datos de usuario se envían mediante un método HTTP POST al enrutador. Este se encarga de invocar al inicio de sesión en el controlador.

Una vez el controlador es llamado, toma la contraseña y la encripta también en SHA1 antes de llamar al modelo. Este último verifica si el usuario existe en la base, y si la contraseña encriptada es la misma que existe en la base de datos.

En caso afirmativo, el nombre del usuario es guardado en una variable de sesión que se mantendrá hasta que el usuario pulse “cerrar sesión”. En caso contrario devolverá un false.

Si los datos fueron correctos, el controlador llamara de nuevo al modelo para pedir los datos del usuario, los cuales se cargan en variables temporales, pues serán necesarios para su perfil, y su avance en el juego. Posteriormente invoca a los fragmentos de la vista que crean la GUI, y los mezclará con los datos obtenidos del modelo, para crear la interfaz del usuario logueado.

Si los datos fueron incorrectos, se cargará de nuevo la vista de página principal, donde el usuario puede intentar iniciar sesión de nuevo, pero cargará además una ventana modal JavaScript dinámica que le indica al usuario que sus datos fueron incorrectos. Ante la poca personalización de las ventanas modales propias de JavaScript, se han desarrollado unas ventanas modales propias personalizables para utilizar en el juego, que posiblemente sean liberadas como código libre una vez finalizado el proyecto. Su desarrollo fue independiente al del juego, pero su integración ha sido importante para el progreso del mismo.

Finalmente, cabe destacar algunos aspectos de seguridad. El método de encriptamiento utilizado, SHA1 permite generar una cadena de 40 caracteres de apariencia aleatoria partiendo de cualquier contraseña. El proceso es de un solo camino: Es posible llegar a la cadena generada a partir de la contraseña original, pero no viceversa. De hecho, para romper este algoritmo es necesario la fuerza bruta, o bien, complejos algoritmos matemáticos de complejidad n^{68} . Esto implica que utilizando contraseñas seguras, un equipo puede tardar años para encontrar la contraseña original. Si alguien no relacionado al proyecto, tuviera acceso a la base de datos con las contraseñas no podría hacer nada con ellas. Como se ha dicho, es casi imposible su desencriptación, y si intentara acceder con la frase encriptada, esta se reencryptaría dando una cadena totalmente distinta, por lo cual no podría acceder. De esta manera se garantiza la seguridad de los datos de los usuarios.



Juego

El juego se realiza utilizando las bondades de HTML5 con su lienzo de dibujo “canvas”, y JavaScript. Todo el juego se lleva a cabo en el cliente. Por su parte, los datos de usuario están guardados en el servidor, y para acceder a ellos es necesario un lenguaje de servidor que en este caso ha sido PHP. Por obvias razones, es necesario mantener una comunicación entre el juego en el cliente y los datos en el servidor: Actualizar puntajes, actualizar medallas, guardar el nivel actual, entre otros.

Cuando el juego se inicia, en el servidor se crea dinámicamente un objeto JSON que contiene toda la información del usuario que se requiere durante el juego. En el cliente, utilizando Ajax se carga este objeto JSON y se guarda como una variable temporal que se mantiene durante la ejecución del juego.

Cuando el usuario supera un subnivel, deben actualizarse su puntaje, habilitarse el nivel siguiente si estaba bloqueado, y en caso de ser el último subnivel de un nivel, habilitar también el siguiente subnivel. Estos datos están guardados en la base de datos. De nuevo vía Ajax los datos se actualizan primero en el objeto JSON inicial, y luego se reenvían al enrutador mediante un método POST de manera asíncronica para finalmente ser actualizados en la base de datos también. De esta manera se garantiza que el progreso de un usuario se conserve aun después de recargar la página, cerrar sesión, o cargar desde otro equipo diferente.

Aunque en esta sección se habla indistintamente de cliente y servidor, la arquitectura de la aplicación sigue basándose en MVC. Las acciones del servidor son realizadas por el servidor desde el enrutador, mientras las del cliente están en el juego, que ha sido cargado desde las vistas con anterioridad por el controlador.

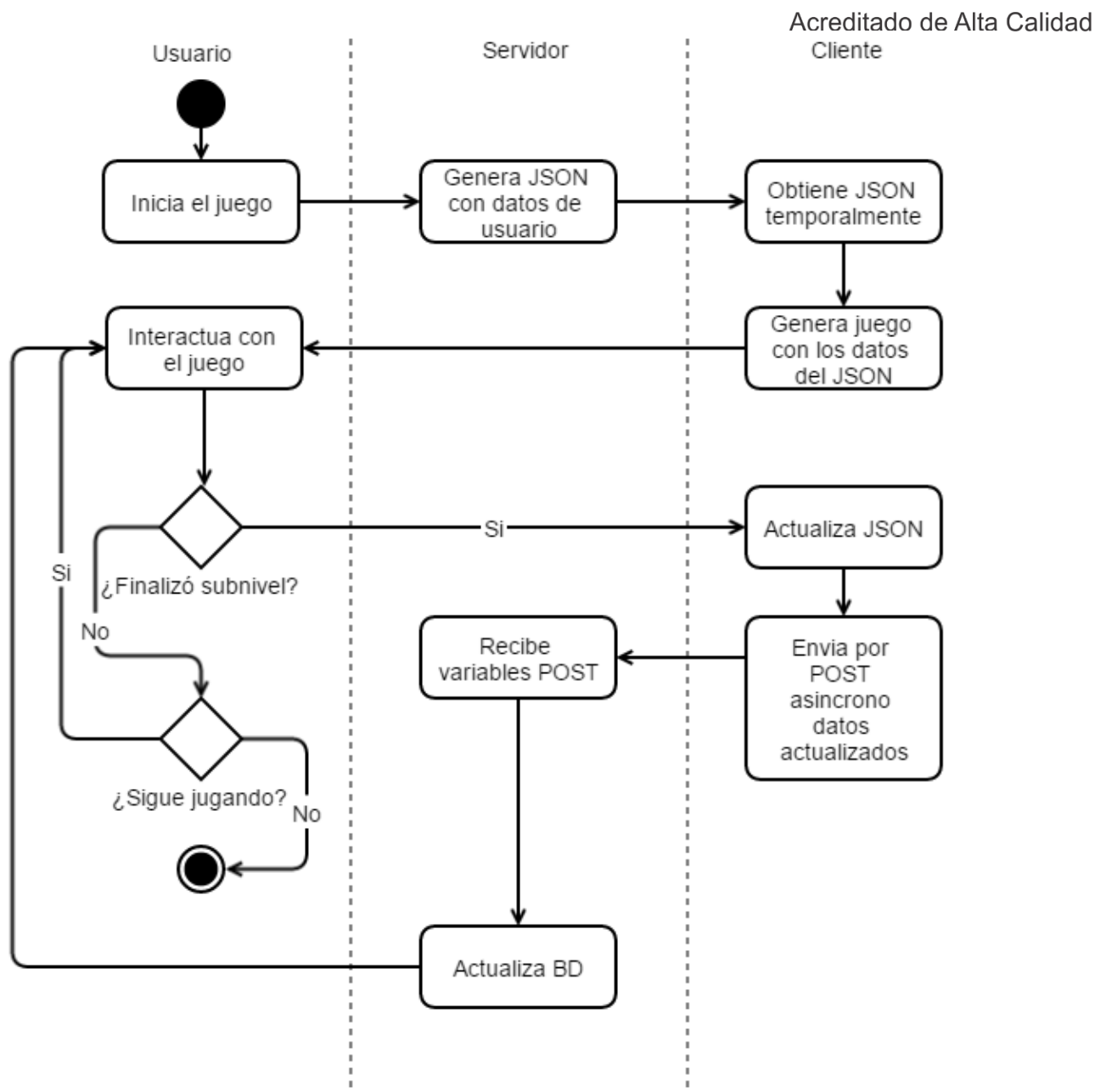


Figura 17. Diagrama de actividades de la interacción del juego



Implementación

Clases utilizadas en el diseño de la solución

Existen superclases en las cuales se implementan los métodos utilitarios que pueden ser tomados por las clases hijas. En el controlador existe la clase controlador, con los métodos necesarios para el manejo de la página.

En el modelo existe también la superclase modelo, la cual contiene los métodos para realizar conexión, consulta y desconexión a la base de datos. De ella heredan las clases de usuario, docente y curso, con los métodos específicos de cada uno de ellos.

La clase del usuario permite registrar nuevos usuarios, actualizar sus datos, retornar los valores del usuario en un array o retornar valores individuales (necesarios para el juego) y modificar su nivel subnivel y puntaje, lo cual se utilizará a medida que el usuario avance en el juego.

La clase del docente contiene funciones también para actualizar sus datos, pero además puede ver datos de los cursos que tiene a su cargo.

La clase del curso contiene los métodos para insertar estudiantes en un curso, y ver los datos de los estudiantes a cargo. Por lo tanto el docente verá datos de sus alumnos a través de la clase curso.

Es importante resaltar que estas clases se comunican, pero cada una tiene su labor específica dependiendo de si corresponde al modelo, o al controlador. Una clase del controlador no puede acceder a base de datos. A su vez una clase del modelo, no puede llamar vistas, ni mostrar datos en pantalla. Para esto deben hacerse llamados de una capa a la otra siguiendo el flujo normal del juego.



REFERENCIAS BIBLIOGRÁFICAS

- Assaf Marron, Gera Weiss, and Guy Wiener. 2012. A decentralized approach for programming interactive applications with JavaScript and blockly. In *Proceedings of the 2nd edition on Programming systems, languages and applications based on actors, agents, and decentralized control abstractions* (AGERE! 2012). ACM, New York, NY, USA, 59-70.
<http://doi.acm.org/10.1145/2414639.2414648>
- Deepak Kumar. 2014. Digital playgrounds for early computing education. *ACM Inroads* 5, 1 (March 2014), 20-21.
<http://doi.acm.org/10.1145/2568195.2568200>
- Katerina Perdikuri. 2014. Students' Experiences from the use of MIT App Inventor in classroom. In *Proceedings of the 18th Panhellenic Conference on Informatics* (PCI '14). ACM, New York, NY, USA, , Article 41 , 6 pages.
<http://doi.acm.org/10.1145/2645791.2645835>
- Ben Gibson and Tim Bell. 2013. Evaluation of games for teaching computer science. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education* (WiPSE '13). ACM, New York, NY, USA, 51-60.
<http://doi.acm.org/10.1145/2532748.2532751>
- Rebecca Vivian, Katrina Falkner, and Claudia Szabo. 2014. Can everybody learn to code?: computer science community perceptions about learning the fundamentals of programming. In *Proceedings of the 14th Koli Calling International Conference on Computing Education Research* (Koli Calling '14). ACM, New York, NY, USA, 41-50.
<http://doi.acm.org/10.1145/2674683.2674695>
- The magic of coding. (2014, Oct 12). University Wire Retrieved from
<http://search.proquest.com/docview/1610681662?accountid=43636>
- Mlot, S. (2013). Play-i robots teach kids to code. PCmag.Com, Retrieved from
<http://search.proquest.com/docview/1529333948?accountid=43636>
- The learning code. (2013, Nov 26). Mint Retrieved from
<http://search.proquest.com/docview/1461782243?accountid=43636>
- Grupo Lifelong Kindergarten, Laboratorio de Medios del MIT. (2003). *Scratch*.
Obtenido de <http://scratch.mit.edu/>
- Alice - Carnegie Mellon University. (2008). Obtenido de Carnegie Mellon University:
<http://www.alice.org/index.php>
- Andes, U. d. (s.f.). *Proyecto CUPi2*. Obtenido de <http://cupi2.uniandes.edu.co/sitio/>



CheckiO. (s.f.). *CheckiO is the game for coders*. Obtenido de <http://www.checkio.org/>

Duarte, E. (07 de Junio de 2012). *Capacity Academy - ¿Por Qué Aprender A Programar?* Obtenido de Capacity Academy - Blog:
<http://blog.capacityacademy.com/2012/05/25/por-que-aprender-a-programar/>

FightCode. (s.f.). Obtenido de <http://fightcodegame.com/>

García, J. C. (2013). *Scratch en la Educación Escolar*.

López, M. F., del Olmo, P., & Reyes, C. (17 de Octubre de 2013). *La programación lúdica como estrategia de articulación entre niveles*. Obtenido de Universidad de Salta, Argentina:
http://sedici.unlp.edu.ar/bitstream/handle/10915/27559/Documento_completo.pdf?sequence=1

RutaN. (s.f.). *RutaN Medellín Centro de Innovación y Negocios*. Obtenido de <http://rutanmedellin.org/index.php/es>

Univerdad de los Andes. (s.f.). *Proyecto CUPi2*. Obtenido de <http://cupi2.uniandes.edu.co/sitio/>

Zuckerberg, M. (s.f.). *Anybody can learn*. Obtenido de Code.org: Code.org