

1.2. Justificación del proyecto y objetivos

- **¿Por qué este proyecto?**

Esta temática me permitirá desarrollar todo lo que hemos aprendido en el curso, ya que el propósito de este proyecto sería abarcar tanto la parte Frontend a nivel de diseño interactivo y entendible para el público al que está dirigido, como la Backend para la gestión de niveles y seguridad.

De manera lógica, es una buena manera de demostrar la programación y flujo que lleva cualquier plataforma web. A su vez, como la dinámica de uso se basa en un funcionamiento parecido al de un videojuego, la lógica a la hora de programar puede tomar más funcionalidades y hacerla más compleja.

También permite implementar una seguridad y control de acceso, ya que esta web requerirá la creación y confirmación por usuario. El paso de este usuario se deberá controlar y delimitar en ciertas zonas de la web.

- **Referencias tomadas del mercado actual**

Actualmente, disponemos de una gran cantidad de herramientas en el mercado para aprender sobre la programación y, de manera autodidacta, llegar a tener una buena base de conocimientos en este campo. Para todos los niveles, desde el más básico hasta para formaciones más profesionales.

Encontramos un amplio catálogo de cursos disponibles en múltiples plataformas como OpenWebinars, Domestika, Platzi, etc. Estas webs están dirigidas en su mayoría para personas adultas, aunque sea de un nivel básico no es lo mismo enseñar a infantes o adolescentes a programar.

Para este aprendizaje en edades tempranas se tiene que contemplar que la edad media en la que los pensamientos abstractos no son tan complejos es a partir de los 10 a los 15 años ya que, aunque se puede enseñar de una manera sencilla instrucciones de uso lógicas, el cerebro no está aún lo suficientemente desarrollado para comprenderlas.

Por ello, lo más normal es que las webs de aprendizaje para infantes/adolescentes sean una combinación de lógicas de programación lo más sencillas posibles combinando estos ejercicios con juegos y estéticas parecidas a la de los videojuegos.

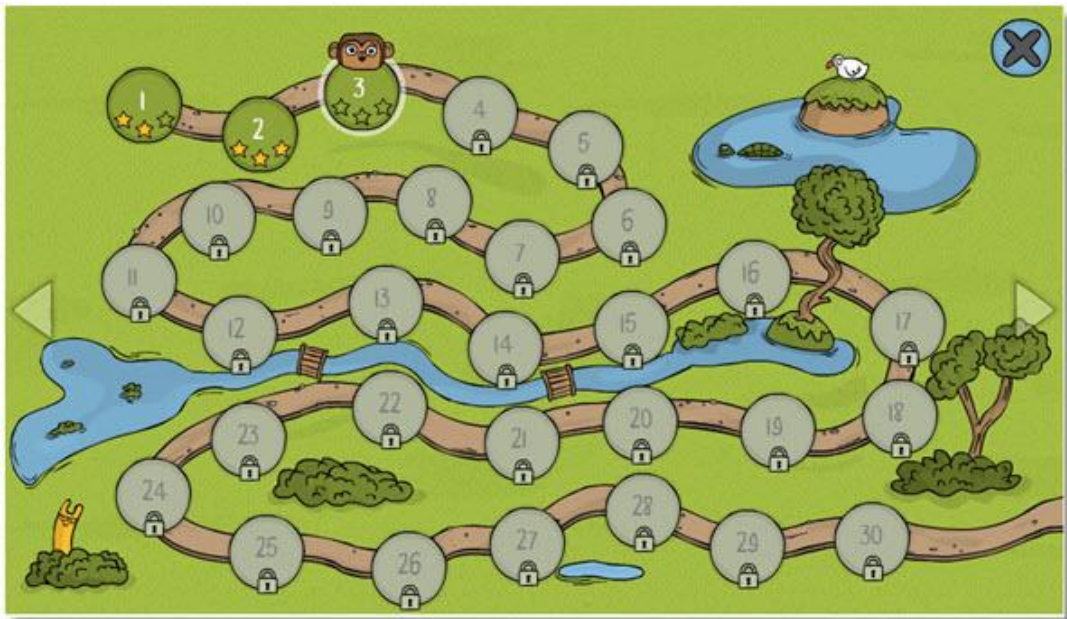
Aprender a programar estimula la creatividad, enseña a trabajar en equipo y el desarrollo de un pensamiento lógico. La programación es una habilidad fundamental a día de hoy, es muy interesante que desde pequeños sea adquirida.

Unos ejemplos de webs que llevan estas dinámicas de enseñanza son las siguientes:

→ Code Monkey



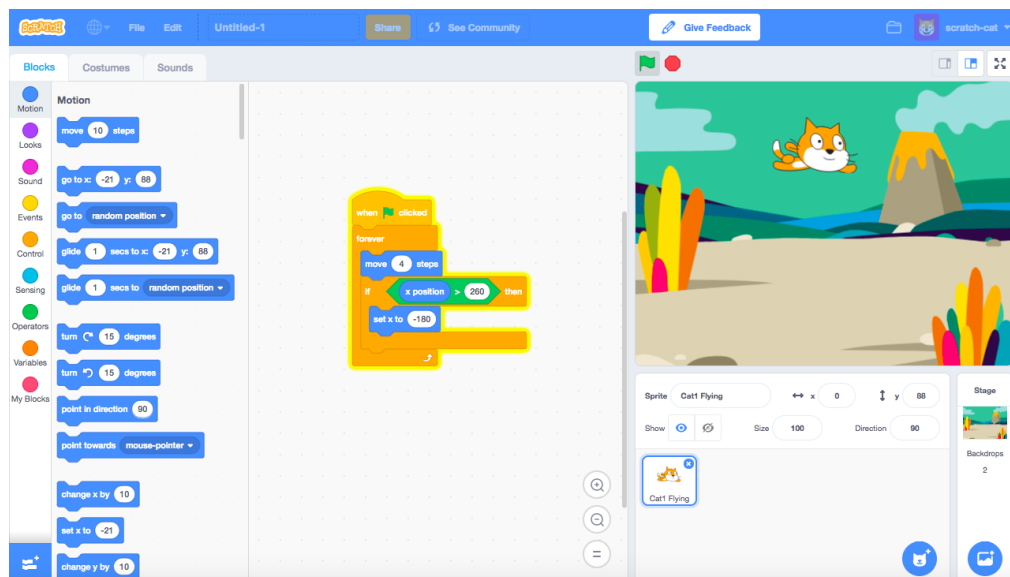
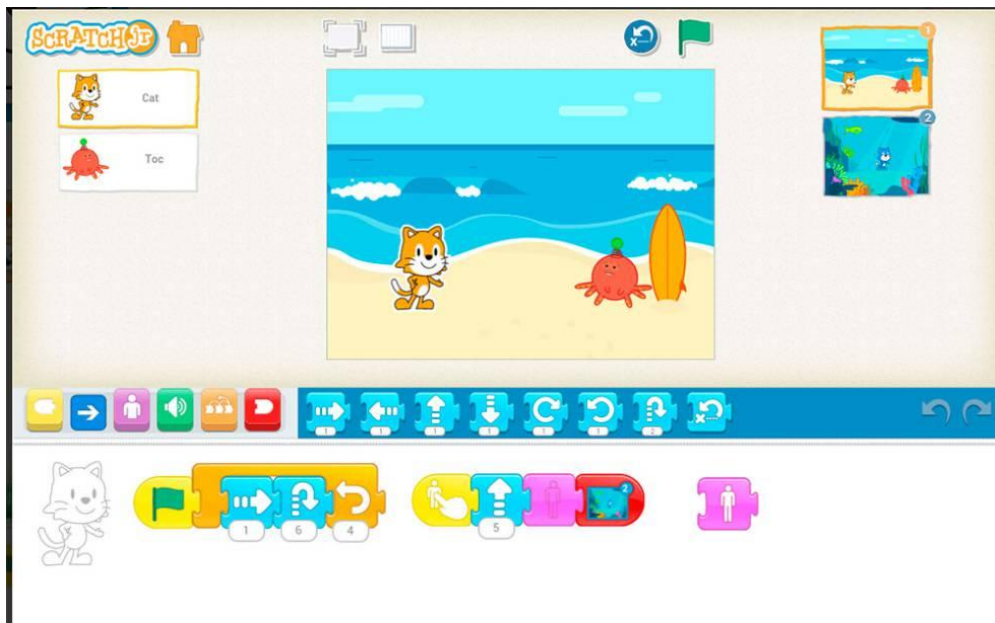
Esta web nos muestra la lógica que se lleva a cabo cuando programamos de una manera sencilla y muy enfocada a la dinámica de los videojuegos. Es una buena manera de enseñar a los infantes a solucionar problemas con lógica básica, pero no es una programación como tal.



→ Scratch

SCRATCH

Una de las más conocidas, dispone de un amplio catálogo de ejemplos, ideas y ejercicios de programación más enfocados a la realidad de esta. Los ejercicios se llevan a cabo mediante la lógica de bloques, estos ayudan a comprender las interacciones y posibilidades que tienen disponibles a la hora de crear una aplicación.



→ Coding Giants



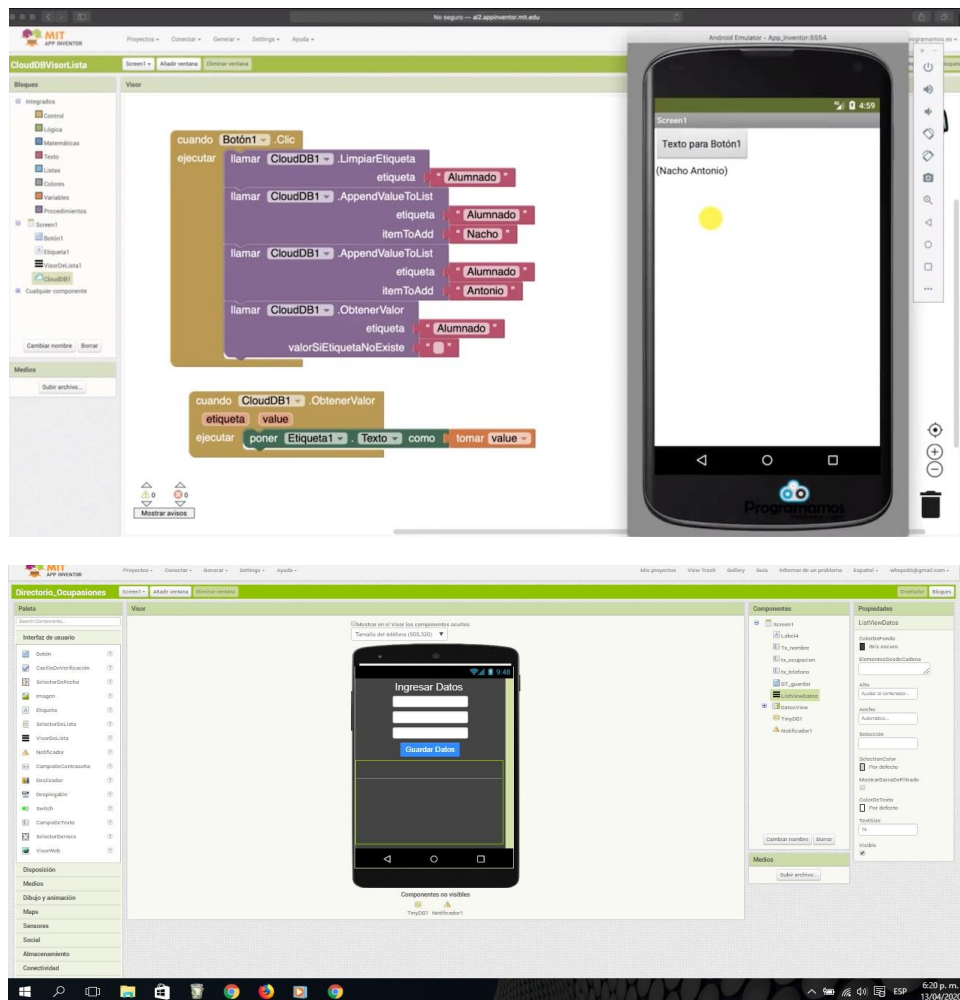
Esta web contiene un catálogo de cursos individuales para todas las edades. Muchos de estos están basados en el diseño y desarrollo de videojuegos, hay otros más teóricos y puramente prácticos a lo que es la realidad, una buena combinación de métodos para las edades a las que están dirigidos.



→ MIT App Inventor



Nos permite crear aplicaciones móviles desde cero, con lógicas de programación en bloque muy similares a lo que puede ser la programación real al igual que Scratch.



● Objetivos

A ser posible, la idea sería dar más peso al Frontend, ya que el funcionamiento de la aplicación en sí puede resultar más sencilla en su planteamiento.

Al ser una web para niños y adolescentes, es lógico que el diseño sea más llamativo, con mayor colorido pero sin dejar atrás una buena estructura de la plataforma, que sea intuitiva y comprensible para ellos. También esto puede llevar mayor estudio y planeación, ya que es una gran parte de mercadotecnia a la hora de exponer el proyecto. En definitiva, la mayor parte del estudio estaría enfocado en la interfaz del usuario para una experiencia más fluida, sencilla y divertida para el sector al que está enfocado esta idea.

Para el desarrollo de esta aplicación, me gustaría trabajar con la herramienta de framework de Spring ya que será la manera más útil de manejar el código Java y mejorará el rendimiento de

la aplicación. También me gustaría implementar el uso de JPA como complemento de Hibernate, ya que es un framework que he encontrado muy útil en el curso. Como base de datos me gustaría probar la opción o de bien hacerlo con Mongo o PostgreSQL, dependerá de cómo se adapte JPA a estas.

Por la parte del Frontend, aparte obviamente del uso de JavaScript y Bootstrap, es donde me gustaría implementar una herramienta no estudiada en el curso, algunas opciones que tengo sería trabajar algo más en profundidad el compilador de Sass y JQuery.

Si fuera posible, también me gustaría estudiar el diseño UX/UI mediante la herramienta de Adobe XD para crear el prototipo de la aplicación y el diseño responsive en ella.

- **Perfil del público**

El público al que estará enfocado será para una edad de 11 años en adelante, ya que después de haber leído sobre cuál es la mejor edad para empezar este aprendizaje, creo que podría ser un buen punto de partida. Ya que a esta edad se empieza a tener un pensamiento más desarrollado y un entendimiento del entorno, la estética de la web puede ser más dinámica sin perder la sencillez.

Como la web tiene que acercarse lo más posible a la realidad, los ejercicios y dinámicas de uso pueden ser más variadas al contrario de las que tendrían que ser para niños más pequeños. Tampoco puede ser que la web se convierta en un videojuego por completo, ya que eso complicaría el proyecto y no sería viable para el tiempo que hay disponible. Los test que se presentan pueden ser entonces una combinación de preguntas teóricas y ejercicios visuales. También sería posible que, al ser una aplicación dirigida a estudiantes, los profesores a cargo de estos puedan tener un historial de sus alumnos y ver su desarrollo según avancen con las lecciones en la web. Al no ser el público principal al que va dirigido no habría una necesidad de cambiar la estética de la plataforma, simplemente es una adición a la estructura principal, que es la que está completamente dirigida a los alumnos y añadiría un login distinto entre alumnos y profesores.