ClimbCode es el resultado de fusionar la enseñanza y la programación, siendo una plataforma web cuyo objetivo es servir de apoyo a la enseñanza de las asignaturas impartidas en institutos y academias mediante el uso de ejercicios estructurados con programación.

Estos ejercicios no son más que una alternativa interactiva a los ejercicios en papel, a los cuales se les ha aplicado programación para poder dar lugar a este nuevo tipo de ejercicio. Este contenido sería el principal servicio ofrecido por la plataforma.

Esto hace surgir la duda de si es necesario saber programar para poder usar el contenido de la plataforma.

La respuesta es que no es para nada necesario, ya que los ejercicios están pensados para hacerse de forma interactiva, y tanto los alumnos como los profesores pueden hacer uso de ellos sin ser necesarios conocimientos de programación en absoluto.

Entonces, ¿De dónde vienen los ejercicios? La respuesta es simple. Todo el contenido de la plataforma lo generan programadores independientes, los cuales con sus conocimientos pueden ir creando los diferentes ejercicios interactivos. Estos ejercicios serán revisados antes de implantarlos en la plataforma para que puedan estar a disposición de los otros usuarios.

Aquí es donde entran las escuelas, las cuales una vez esté aprobado e incluido el ejercicio en el sistema, pueden adquirir ejercicios para ampliar el catálogo de los que disponen.

Para hacer estas compras, las escuelas podrán visualizar una lista con todos los ejercicios interactivos del sistema, pudiendo elegir los que deseen adquirir de entre ellos y pagando su precio correspondiente.

Una vez adquiridos estos ejercicios, los profesores podrán hacer uso de ellos para apoyar las clases que impartan, igual que los alumnos podrán acceder a ellos para completarlos y usarlos de suplemento en su aprendizaje. Además, los profesores podrán subir y bajar documentación de sus asignaturas a la plataforma, y los alumnos podrán acceder a ella.

Por último, pero no menos importante, el papel de los administradores en el producto consistiría en ir registrando nuevas escuelas que quisiesen entrar en el sistema, para que además éstas pudiesen ir realizando sus labores de gestión de profesores y alumnos. Los administradores también promocionarán los ejercicios que soliciten los programadores con un pago correspondiente por parte de ellos.

Para ilustrar mejor cómo serían los principales procesos del sistema, hemos elaborado una serie de mockups.

En el primero de ellos, vemos desde el punto de vista del alumno la página principal de la plataforma una vez logado en ella. Para poder hacer un ejercicio, accedemos a la pestaña “Mis asignaturas”.

Dentro de ésta, podemos observar todas en las que está matriculado el alumno. Para ver el contenido de una de ellas, bastaría con seleccionar su nombre.

Lo cual nos mostraría la lista de documentos y ejercicios de esa asignatura. Para hacer uno de los ejercicios, de nuevo seleccionamos su nombre.

Esta acción nos llevaría al propio ejercicio, en el que se mostraría el título, su enunciado, un dibujo en el que ir mostrando los resultados del propio ejercicio, y explicaciones sobre cómo se va a proceder para resolverlo, si no viene resuelto ya. En este caso el alumno sólo tendría que ir modificando los valores de la velocidad inicial de la pelota para ver el resultado del ejercicio con el botón de ejecución. Adicionalmente podría añadirse una característica para que el alumno pudiese ver el código subyacente al ejercicio y poder entender su lógica programática.

El siguiente mockup nos muestra lo que vería un programador recién registrado en el sistema. Para poder crear un ejercicio, bastaría con acceder a la sección “Mis Ejercicios”.

Dentro de ella, vemos una lista con todos los ejercicios hechos por el programador, así como algunas de sus características. Para crear un nuevo ejercicio accedemos al botón correspondiente.

Lo cual nos llevaría a una nueva vista en la que podríamos configurar nuestro nuevo ejercicio mediante las herramientas que dispone la plataforma. Una vez tengamos el ejercicio finalizado, lo confirmamos y se añadiría a la lista de ejercicios del programador, a la espera de su comprobación e implantación.

Una vez vistas la idea principal y cómo funciona la plataforma, lo siguiente que nos acucia es saber si esta idea ya está en el mercado. Hemos estudiado a muchos de nuestros posibles competidores, y el resultado siempre es el mismo: Están centrados en la gestión académica y documental, sólo enseñan a programar o sólo dejan hacer ejercicios improvisados y planos, sin un soporte en la programación.

En cuanto a las fuerzas de nuestro proyecto, destacamos que, como hemos visto justo ahora, es diferente a los demás del mercado, pues ninguno realiza nuestras dos funciones a la vez: Gestión documental y ejercicios interactivos. La otra fuerza es que al no ser necesario saber programar para usar la aplicación siendo un profesor o un alumno, ésta es bastante accesible para ellos.

Respecto a las debilidades, nos encontramos con que puede que a pesar de ser útil, nuestro proyecto no encaje bien en la mecánica de aprendizaje que desarrollan algunos profesores. La otra debilidad es que nos resultaría difícil expandirnos a otros nichos de mercado, dado que la aplicación está demasiado centrada en la enseñanza.

Pasando a las oportunidades, las dos principales son que es posible saltar a otra plataforma diferente a los ordenadores, como en el caso de las tabletas, que son bastante usadas hoy en día por los docente, y que es posible añadir bastantes nuevas funcionalidades una vez terminado el proyecto, por lo que podríamos abarcar más de las necesidades de nuestros clientes.

Por último, las amenazas a las que hace frente nuestro proyecto son dos: La posibilidad de que nuestros competidores se hagan con nuestra idea y la implanten en sus productos en caso de ser exitosa, lo que nos dificultaría competir con ellos, y la posibilidad de que por subestimar la dificultad de implantar las funcionalidades del proyecto, éste sufra retrasos.

Para saber si el proyecto es viable, hemos realizado una encuesta a posibles interesados.

En esta encuesta participaron 88 personas, de las cuales el 70% estaban relacionadas con la enseñanza.

Ante la pregunta de si creen que es necesario acercar a los alumnos a la programación dado el auge del perfil informático en el mundo laboral, casi la totalidad de los encuestados respondió que sí.

Respecto a si querían que para lograr esto hubiese una herramienta de apoyo en las escuelas, de nuevo la mayoría respondió afirmativamente.

Ya entrando en si usarían dicha herramienta o no, aunque la respuesta es mayormente afirmativa, sí es verdad que si fuese demasiado compleja preferirían hacerlo con material de apoyo.

Por último, se les preguntó si creían que con esta herramienta los estudiantes se tomarían la educación de manera más amena, para lo cual de nuevo la mayoría respondió afirmativamente.

Una vez vista la aceptación del producto, nos surge la duda de saber cuánto costaría hacer el proyecto en la práctica.

Así pues, en el espacio de tiempo en el que se desarrollan la asignatura y el proyecto y teniendo en cuenta los costes directos, indirectos y las amortizaciones mensuales…

Nos saldría un total de 9250€ mensuales, que multiplicados por la duración y sumándole 4000€ de reserva, darían un total de 41000€.

Pero, ¿cómo recuperar esta inversión?

La primera manera que tendríamos para obtener beneficio sería mediante la venta de licencias a las escuelas. Estas licencias anuales permitirían a las escuelas hacer uso de la plataforma al completo mediante un curso entero, siendo necesario renovarlas cada año para mantener los servicios. Además, junto con la licencia se le incluiría además del acceso a la plataforma una serie de ejercicios iniciales.

Así pues, los criterios para clasificar estas licencias serían el número de usuarios permitidos al cargo de la escuela que contrate la licencia, junto con un número de ejercicios iniciales gratuitos a elegir por el comprador. Esto nos daría como resultado estos tamaños estándar de licencias, con su precio asociado. Además, en caso de no encajar con ninguno de estos tamaños, el cliente siempre podría contactar con nosotros para adaptar la licencia a sus necesidades en cuanto a usuarios.

La segunda manera que tendríamos para obtener beneficio, sería con la venta a las escuelas y academias de los ejercicios interactivos en la plataforma. Los ejercicios interactivos tienen un precio de compra inicial, del cual nosotros nos quedaríamos con un pequeño porcentaje mientras que el autor del mismo recibiría el resto del precio como beneficio para él, lo que además actuaría como motivante para los creadores de contenido de la plataforma.

Como aclaración, todas las transacciones mencionadas anteriormente se harán a través de PayPal, cuyas funcionalidades integraremos en nuestra plataforma.

Como metodología de trabajo, adoptaremos la presentada en la asignatura: SCRUM, concretamente con tres sprints.

En el primero de ellos nos centraremos en elaborar los CRUD básicos del sistema, junto con una primera iteración sobre el sistema de ejercicios interactivos, cuyo objetivo en este sprint sería poder realizar y crear ejercicios, pero no persistirlos.

Así pues, para los dos sprints restantes nos quedaría añadir esta persistencia para terminar el sistema de creación de ejercicios, los sistemas de compra de los mismos, las funcionalidades de la gestión de documentos de las asignaturas y los posibles métodos de búsqueda de ejercicios y correcciones referentes al feedback que vayamos recibiendo.

En cuanto a tecnologías de desarrollo, hemos pensado utilizar Python, Django y PostgreSQL para la parte principal de la aplicación, así como Notebook y JavaScript para la creación y realización de ejercicios. Estas tecnologías son conocidas por el equipo, aunque no son las más usadas por ellos.

En cuanto a la gestión del conocimiento, haremos uso de un repositorio de git, alojado en github, en el que realizar un seguimiento de la evolución del proyecto y sus tareas, así como almacenar las documentación y el conocimiento generados.