

¿Que es ClimbCode?

Opción 1: Centrado en los ejercicios

ClimbCode es una plataforma orientada a apoyar las enseñanzas de otras asignaturas mediante el uso de programación.

Su principal uso sería almacenar todos los contenidos de las asignaturas y ofrecer apoyo a éstas mediante ejercicios interactivos, que muestran los conceptos de las asignaturas mediante programación.

- Almacén de documentación
- Ejercicios proporcionados de la asignatura resueltos mediante programación sencilla

Ventajas: Más simple, nos centramos en los ejercicios

Desventajas: menos funcionalidad. Las escuelas podrían rechazarlo al ser solo para pocas asignaturas

Opción 2: Enseñanza virtual

- Almacén de documentación
- Ejercicios (igual que en 1)
- Ejercicios tipo test, sin programación
- Competiciones
- Entregables

Ventajas: Más funcionalidades: las escuelas podrían validarlo al poder manejar todas las asignaturas. Proporcionamos gamificación.

Desventajas: Más complejo. Menos tiempo para el módulo de ejercicios.

ClimbCode es una plataforma de enseñanza virtual orientada a llevar la programación a todos los ámbitos de la educación.

Surge como respuesta a la creciente tendencia de llevar la programación al currículo escolar, y de una manera de dar soporte informático a la enseñanza.

De esta forma, unimos estos dos conceptos: el apoyo a la enseñanza con medios informáticos y los conocimientos de programación para resolver problemas, con un medio divertido y fácil de usar por medio de la gamificación, de ahí nuestra adorable mascota.

El objetivo es hacer un entorno interactivo y divertido por medio de la gamificación, haciendo la enseñanza de las asignaturas más amenas por medio de la programación.

Si bien el enfoque de ClimbCode es aplicar la programación a las asignaturas, no se limita solo a ello, ClimbCode da soporte a crear ejercicios, entregables, evaluaciones, etc. por lo que puede servir de soporte a toda la enseñanza.

¿A quién está destinado?

Opción 1

Está destinado a los alumnos de institutos y academias de 3º de la ESO en adelante en cuanto al uso de los ejercicios y la documentación y a programadores independientes en cuanto a la creación de ejercicios.

Opción 2

Está destinado a toda la enseñanza, cualquiera que quiera mejorar la forma de enseñar puede hacer uso de ClimbCode para...

¿Quién crea el contenido?

ClimbCode proporciona una serie de ejemplos ya creados para asignaturas como biología, matemáticas, física... Pero cualquiera puede presentar propuestas de ejercicios que se llevarían a todas las escuelas que implementarán ClimbCode, además de que cada profesor puede crear sus propios ejercicios. Si tu ejercicio es aceptado recibirás una compensación económica. Para que el ejercicio sea aceptado debe incluir además su resolución.

LOS PROFESORES NO PUEDEN CREAR EJERCICIOS

Opción 1: Cualquiera propone ejercicios, validamos y pago único al autor.

Ventajas: Controlamos que ejercicios entran. No tenemos que controlar el éxito

Desventajas: Probablemente menos incentivo para el autor. Costoso

Opción 2: Cualquiera propone ejercicios, validamos, se paga en función del uso por parte de escuelas.

Ventajas: Controlamos los ejercicios. Mejor incentivo para el autor

Desventajas: Costoso. Hay que controlar el éxito del ejercicio

Opción 3: Cualquiera propone ejercicios, no validamos, se paga en función del uso por parte de las escuelas, los más usados aparecen antes.

Ventajas: No hay que controlar los ejercicios, menos costoso.

Desventajas: El profesor tiene que hacer un filtrado. Hay que controlar el éxito

¿Que características tiene?

Opción 1

- Subida de la documentación de la asignatura.
- Ejecución de ejercicios.
- Añadido de ejercicios en la lista de los de una escuela.
- Creación de ejercicios por parte de programadores independientes.

Opción 2

ClimbCode es una plataforma de enseñanza virtual, en la que gestionar las asignaturas impartidas y la documentación de cada una, así como crear ejercicios para que resuelvan los alumnos.

A continuación se enumeran:

- Gestión de documentación de la(s) asignatura(s)
- Creación de ejercicios, estos pueden ser de tipo test, rellenar respuesta, de programación... Se proporcionan una serie de ejercicios ya planteados que los

profesores pueden incorporar a la asignatura, además de que pueden crear sus propios enunciados si lo desean

- Creación de competiciones: los alumnos deberán responder en tiempo real a una serie de preguntas en el tiempo establecido (Por pregunta o por ejercicio completo) ganando puntos por respuesta correcta según el tiempo que hayan tardado.
- Sistema de entregables con evaluación: ya sean ejercicios o trabajos entregables.

¿Cuánto cuesta?

Opción Antigua:

Nosotros damos distintos tipos de licencias que dan soporte a un número limitado de usuarios y asignaturas que dar de alta en el sistema, aunque el número es flexible, si fuese necesario personalizar una licencia contacte con nosotros. *(Deberíamos ofrecer packs de alumnos, no negociar el número exacto, en plan 5 alumnos más -> 20€ al mes, hay que ser flexible pero esto no es un bazar de Marruecos)*

Licencias de las que disponemos:

Licencia A: 100 usuarios(alumnos + profesores), 1000€/año *(ejemplo)*

Opción 1: Pago anual/mensual por usuarios, todos los ejercicios gratis. **NO:** Remuneración a programadores difícil.

Opción 2: Pago anual (opcion1, según usuarios) + packs de ejercicios incluidos, compras más ejercicios. Ofertas para X ejercicios comprados a la vez? Las asignaturas vienen dadas por los ejercicios que dispones. **Esta**

Opción 3: Sin pago anual, pack inicial, pagas solo por nuevos ejercicios. **NO**

Opción 4: Compra de la aplicación, sin pago anual y solo ejercicios. **NO**

Compra de ejercicios:

Ventajas: Remuneración. Personalizable

Desventajas: Gestión del pago (para nosotros y para la escuela). Los profesores no podrán crear ejercicios, por lo que no puede ampliarse a otras asignaturas.

Para los costes del proyecto, añadir proyección ROI (Estimación optimista,pesimista,media, beneficios,*break even*,)

¿Cómo son los ejercicios?

ClimbCode dispone de ejercicios resueltos que expliquen cómo se resuelven, ejercicios guiados que llevarán paso por paso a resolver el problema, rellenando las partes que falten, y ejercicios de esta forma los alumnos entienden cómo se resuelve y lo hacen de una forma más interactiva.

Los ejercicios tienen una asignatura. Los ejercicios aparecerán ordenados por esas asignaturas.

- ¿Asignatura? ¿O asignatura + curso?
 - Opción 1: “Asignatura X” - “Curso Y” (dos tipos: Asignatura, Curso)
 - Opción 2: “Asignatura Y del curso X” (un tipo: Asignatura).

- Ejercicio resuelto: ejercicio explicado de como resolver el problema paso a paso.
- Ejercicio guiado/parametrizado: ejercicio planteado en la que solo hay que rellenar las partes que faltan/añadir parámetros para que salga el resultado.
- Ejercicio libre? solo enunciado.

Lenguajes de los ejercicios. Estudiarlos a ellos, sus tecnologías y sus riesgos:

- JavaScript
- Python

MOCKUPS DE EJERCICIOS GUIADOS Y HECHOS. TAMBIÉN DE CREACIÓN DE EJERCICIOS

Ejemplo de ejercicio HECHO (lo que hay detrás del igual seria rellenable por el alumno en uno GUIADO):

Se lanza una pelota con una velocidad de 20m/s a una altura de 10m rellena los huecos para simular el lanzamiento:

En primer lugar apuntamos los datos del problema: Comenzamos indicando la velocidad inicial de la pelota en el eje X y el eje Y:

var velocidad_x_pelota = 20

var velocidad_y_pelota = 9.8*t

A continuación, su posición inicial:

posicion_inicial_x_pelota = 0

posicion_inicial_y_pelota = 10

Ahora, aplicamos las fórmulas TAL y TAL para ver las posiciones finales de la pelota en el eje X y el eje Y:

posicion_x(t) = posicion_inicial_x_pelota + velocidad_x_pelota * t

posicion_y(t) = posicion_inicial_y_pelota - velocidad_y_pelota*t

Y ahora, dibujamos el resultado:

dibujarCirculo(posicion_x(t), posicion_y(t))

¿Cómo es el MVP?

- MVP:
 - Administrador: Dar de alta escuela
 - Escuela: Dar de alta usuarios, comprar ejercicios (gestión interna: profesor envía URL o le dice al "director" que compre el ejercicio tal porque le interesa).
 - Profesor: Previsualizar, para ejecutar ejercicios. Subir y bajar documentación (temarios propios).
 - Alumnos: Ejecutar ejercicios y descargar documentación.
 - Programador: Crear y subir ejercicio categorizados (un select al crearlo) en una asignatura concreta.

¿Cómo organizamos los Sprints?

Opción 1

- Plan (sugiero)

- Sprint 1 :
 - Proyecto base
 - Crud Escuela, Usuarios, Asignaturas, Algún ejemplo de ejercicio
 - *Framework* de ejecución de código en ejercicios
- Sprint 2
 - CRUD Ejercicio, competición, entregable
 - Más ejercicios
- Sprint 3
 - Mejoras

Opción2

- Sprint 1:
 - Proyecto Base
 - CRUD escuela, usuarios, programadores
- Sprint 2:
 - Creación, validación y descarga de ejercicios
 - Ejecución de ejercicios
- Sprint 3:
 - Mejoras

Competidores

- Comparativa con competidores en una tabla.

DAFO

- DAFO:
 - Separadas en función de las características:
 - Plataformas de cursos de programación (Udemy, CodeAcademy...)
 - Plataformas de enseñanza (Blackboard, Moodle, ...)
 - Enseñanza de programación a niños (No me acuerdo que había, Microsoft tenía una)
 - <https://www.microsoft.com/en-us/education/training-and-events/teach-students-to-code/default.aspx>

TAREAS:

- Mockups (Todos juntos)
- Ejemplos de ejercicios (Asignatura + tipo de ejercicio)