

¿Que es ClimbCode?

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

¿A quién está destinado?

Está destinado a los alumnos de institutos y academias con enseñanzas de secundaria y bachillerato (porque hay más variedad de asignaturas) en adelante en cuanto al uso de los ejercicios y la documentación y a programadores independientes en cuanto a la creación de ejercicios.

¿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

Cualquiera (programador) propone ejercicios, no validamos, se paga en función del uso por parte de las escuelas (compran ejercicios aparte del nº que pueda llevar la licencia), 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?

- 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.

¿Cuánto cuesta?

Pago anual según usuarios + packs de ejercicios incluidos, compras más ejercicios a parte.
Ofertas para X ejercicios comprados a la vez? Las asignaturas vienen dadas por los ejercicios que dispones.

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 Y del curso X” (un tipo: Asignatura).
 - Ej: Matemáticas 1º ESO ; Ciudadanía 3º ESO
- Ejercicio parametrizado: ejercicio planteado en la que solo hay que rellenar las partes que faltan/añadir parámetros para que salga el resultado. Puede venir con los parámetros rellenos o vacíos.

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 RESUELTO (lo que hay detrás del igual sería 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? Casos de Uso

- MVP:
 - Administrador:
 - Gestionar alta escuelas.
 - Crear cuenta.
 - Gestionar licencias.
 - CRUD Asignaturas.
 - Listar ejercicios:
 - Eliminar ejercicios.
 - Escuela:
 - CRUD Usuarios: profesores y alumnos.
 - Asociar alumnos con asignaturas.
 - Gestión ejercicios:
 - Listar ejercicios. Sin ejecución: solo descripción y detalle.
 - Filtrado (por nivel, asignatura).
 - Buscador.
 - Comprar ejercicios (gestión interna: profesor envía URL o le dice al "director" que compre el ejercicio tal porque le interesa). De uno en uno -> Vista de pago sin carrito.
 - Gestión contenido asignaturas (esto es lo que suben los profesores).
 - Profesor:
 - Listar ejercicios como la escuela. Sin ejecución: solo descripción y detalle.
 - Filtrado (por nivel, asignatura).
 - Buscador.
 - Previsualizar y ejecutar ejercicios.(Ejercicio comprado)
 - Gestión documentos:

- Subir y bajar documentación (temarios propios).
- Eliminar.
- *¿Control de los alumnos que realizan los ejercicios? Dependiendo carga de trabajo.*

NOTA: Profesores tienen acceso a todas las asignaturas (ejercicios de asignaturas) de la escuela.

- Alumnos:
 - Descargar documentación.
 - Ejecutar ejercicios.

NOTA: ¿Alumnos pueden navegar por todos los ejercicios? No. Solo los de sus asignaturas. La escuela asocia al alumno a unas asignaturas (Ej: “Matemáticas de 1º ESO, Ciudadanía 2º...”) y solo verá los ejercicios de ellas. Estas asignaturas relacionadas con el alumnos son las mismas que las asociadas con los ejercicios.

- Programador:
 - Listar ejercicios. Sin ejecución: solo descripción y detalle.
 - Filtrado.
 - Buscador.
 - Crear ejercicios categorizados (un select al crearlo) en una asignatura concreta (Ej: Matemáticas de 1º ESO). Asignarle nivel (básico, intermedio, avanzado) en otro select.
 - Ver sus ejercicios. Podrá ver el nº de ventas.
 - Ver ingresos/ventas: cada compra se le acumula y en X fecha se realiza el ingreso (y contador a 0 €).
 - Editar perfil.
 - Promocionar ejercicio pagando X cantidad al sistema (nosotros).
 - Posible balance con los ingresos actuales y en caso de no tener fondos, pago a parte.

NOTA: Habrá que definir pantalla de ejercicios promocionados (o en la pantalla de inicio del sistema).

¿Cómo son las licencias que vendemos a las escuelas? ¿Y el precio unitario de cada venta de ejercicio? ¿Precio por promoción?

Los paquetes tipo son:

- Básico: 60 usuarios + 20 Ejercicios, precio 200€/anual
- Medio: 200 usuarios + 50 Ejercicios, precio 550€/anual
- Grande: 500 usuarios + 75 Ejercicios, precio 700€/anual
- Contacto para usuarios extras.

Plan de financiación:

- Básico: $200/12 = 16,66€/mes$
- Medio: $550/12 = 45,83€/mes$
- Grande: $700/12 = 58,33€/mes$

Precio por ejercicio: suponemos estimación de 2-3 horas en la realización del ejercicio por parte del programador: 4 € por ejercicio y el sistema se llevaría 1€.

Precio de promocionar un ejercicio: 1€ cada 2 meses.

COSTES

Costes del proyecto

- 4 meses de desarrollo y 6 meses de mantenimiento:
- 10 horas semanales de trabajo. (En un trabajo real deberían ser 40h/mes)

Costes directos:

- Project Manager: Alejandro Román - 27,19 €/h x 40 h/mes = 787 €/mes
- Business Analyst: Álvaro Domínguez - 16,01 €/h x 40 h/mes = 640,40 €/mes
- Ingenieros Software: El resto. - 11,47 €/h x 40 h/mes = 458,80 €/mes x 5 personas = 2294 €/mes

Coste directo = 3721,4 €/mes

- Amortización de equipos: 7 equipos * 1.000€/ordenador = 7000€

5 años de amortización : $7.000/5 = 1.400€/anual$, $1400/12 = 116,66€/mes$ (Antes en 3 años - 194,44)

$116,66*7 = 816,62€$

$(116,66*4 meses)*7 personas = 3266,48€$ (los 4 meses)

Coste amortización = 816,62€/mes

Costes indirectos:

- Hosting: 5.75€/mes (Pack Hobby Heroku)

20.5€/mes (Pack estándar mínimo Heroku) (Aplicación)

100€/mes Almacenamiento/escuela (En principio no lo necesitamos)

Total hosting = 20,5€/mes (en principio)

- Espacio co-working: 170 € x 7 personas x 4 meses = 4760 € (1190€/mes)

Plan de contingencia:

- 4000€ de reserva

Total/mes = $3721,4+816,62+20,5 + 1190 = 5748,12€/mes$

Para 4 meses = $4540,52€*4 + 4.000 = 18162.08 + 4000 + 4760 = 26922,08$

¿Cómo organizamos los Sprints?

- Plan (sugiero)
 - Sprint 1 :
 - Proyecto base.
 - CRUD básico asignaturas: listado, alta, edición, eliminación.
 - CRUD básico escuelas: listado, alta, edición, eliminación.
 - CRUD básico usuarios (profesores y alumnos): alta, edición.

- *Notebook v1:*
 - *Programador: Que aparezca el notebook en la vista de creación del programador (integrarlo solo). Persistencia en Sprint 2.*
 - *Usuario (Alumno/Profesor): Que aparezca en la vista del alumno. Recuperación del persistente en Sprint 2.*
- Diseño de ejercicios básicos (batería por defecto).
 - Selección de ejercicios (gratis incluidos en el paquete inicial).
- Sprint 2
 - CRUD básicos: documentos (profesores).
 - Notebook v2:
 - Persistencia al crear (programador).
 - Posibles Mejoras.
 - Recuperación de ejercicios persistentes
 - Persistencia de ejercicios básicos (batería por defecto).
 - Descargar documentos (alumnos).
 - Promociones.
 - Compra de ejercicios.
 - Programador: Cobrar los ingresos.
- Sprint 3
 - Mejoras CSS.
 - Filtrados y buscadores.

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:

Martes

- Mockups (El resto)
 - Administrador (Álvaro Sánchez)
 - Escuela (Jesús)
 - Profesor (Ale Román)
 - Alumno (Andrés)

- Programador (Andrés)

Jueves

- Costes + Notebook + Calculadora de horas (Alejandro G.)
- DAFO + Presentación (Álvaro D.)