**MVP**

**ADMINISTRADORES**

* CRUD Escuelas: se le asocia nº asignaturas y usuarios según licencia contratada.

**ESCUELAS**

* Las escuelas piden registro eligiendo licencia con el número de usuarios que desean, a través de un formulario.
* Las escuelas pueden pedir cambiar de licencia.
* Las escuelas crean usuarios (alumnos y profesores) y asignaturas hasta el límite de su licencia, deben poder hacerlo por lotes.
* Las escuelas asignan profesores a las asignaturas.
* Las escuelas asignan alumnos a las asignaturas.

**PROFESORES**

* Los profesores pueden subir documentación a las asignaturas que imparten.
* Los profesores deben poder crear ejercicios (ver la sección de ejercicios)

**ALUMNOS**

* Los alumnos deben poder ver y descargar el contenido de las asignaturas que cursan.
* Los alumnos deben poder completar los ejercicios de las asignaturas que cursan
* Los alumnos deben poder ver el ranking de la asignatura(Esto en verdad los profesores también, mover a usuarios en gral)

**USUARIO ANÓNIMO**

* Ver página de bienvenida
* Ver licencias
* Contacto
* Formulario de petición de alta

**FUNCIONALIDADES EXTRA FUTURAS**

* El profesor puede sumar o restar puntos a un alumno
* El profesor puede copiar contenido entre asignaturas que imparte.
* Sistema de recompensas.

**TECNOLOGÍA**

· Python

· Django

· PostgreSQL

· Bootstrap

**LICENCIAS**

· Las licencias son anuales.

· Las licencias van por número de usuarios y asignaturas.

**SE USARÁ LA OPCIÓN 1**

**Opción 1:** Enfocada a academias, institutos y la Consejería de Educación (Para colegios públicos)

· Los paquetes tipo son:

o Básico: 5 asignatura 60 usuarios, precio 200€/anual

o Medio: 15 asignatura 200 usuarios, precio 550€/anual

o Medio-Grande: 25 asignaturas 500 usuarios, precio 700€/anual

o Grande: 40 asignaturas 800 usuarios, precio 1100€/anual

o A medida: Contactar con el equipo de desarrollo

* Alumno extra: según el tipo de paquete (precio anual) / usuarios
* Asignatura extra: Precio por definir

Cada paquete incluye durante un año:

- Soporte técnico.

- Documentación.

- Actualizaciones del software.

Plan de financiación:

Básico: 200/12 = 16,66€/mes

Medio: 550/12 = 45,83€/mes

Medio-Grande: 700/12 = 58,33€/mes

Grande: 1100/12 = 91,66€/mes

**ROLES DEL EQUIPO Y COSTE**

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)

Plan de contingencia:

· **4000€ de reserva. --- ??**

Total/mes = 3721,4+816,62+20,5 = **4540,52€/mes**

Para 4 meses = 4540,52€\*4 + 4.000 = **18162.08 +** 4000?

**Ejercicios**

1. **Tipos**
   1. Test
      1. Selección múltiple
      2. Selección simple
      3. Combo
      4. Cuadro arrastrable
   2. Programación
   3. Competición (Ver test, posibles: ii e iv)
      1. Preguntas secuenciales, el profesor dispara la siguiente pregunta
      2. Tiempo límite, puntos según velocidad.
2. **Opciones de modificación de los ejercicios**
   1. Intentos límite
   2. Fecha límite
   3. Evaluable por el profesor/Autoevaluable
3. **Creación de ejercicios**
   1. **Test**
      1. Se elige crear el ejercicio de tipo test, se elige el subtipo. Luego se añade el enunciado y se eligen las opciones que van a aparecer. Finalmente se selecciona la correcta y se confirma el ejercicio.
   2. **Programación**
      1. El profesor elige crear el ejercicio de programación. Luego añade el enunciado y el código de respuesta. El código es ejecutable y si es correcto, se puede confirmar el ejercicio. Al corregirlo es el profesor el que lo hace manualmente, aunque se pueda ejecutar lo que vayan haciendo los alumnos.
   3. **Competición**
      1. Igual que el de tipo test, pero añadiendo los parámetros como el tiempo de cada pregunta y los puntos.

**SPRINT 1**

Consiste en desarrollar la funcionalidad mínima que pueda entregarse a los usuarios pilotos con una UI decente.

Se desarrollará:

* Semana 1 (12-Marzo a 16-Marzo):
  + CRUD de las escuelas, usuarios y asignaturas:
  + Página principal
  + Formularios de creación de todos los usuarios
  + Investigación del ejercicio tipo Notebook
* Semana 2 (19-Marzo a 23-Marzo):
  + Subida y almacenamiento de documentación por parte de los profesores
  + Investigación del ejercicio tipo Notebook