# **Tareas a realizar**

**Metodología a aplicar a la hora de empezar una tarea:**

Por cada tarea asignada en el documento:

1. El encargado debe crear la issue en Github.
2. Describir añadiendo a la descripción **una estimación (horas)** de lo que cree que va a tardar.
3. Realizar la tarea.
4. Al finalizar y antes de cerrar la issue, poner la **dedicación real (horas)** de lo que finalmente ha tardado.
5. Rellenar el documento compartido con los problemas encontrados y las soluciones aplicadas. Va en paralelo con lo anterior.
   1. Indicar Issue.
   2. Indicar Problema.
   3. Indicar Soluciones.

<https://docs.google.com/spreadsheets/d/1qGFVwrqfUZ3XMohXejUPHn0WzF91OGpRXjGaXnunSuk/edit#gid=0>

**Reparto de tareas:**

Presentación/Contenido.

* Presentación Power Point. **10 SP**. – Álvaro D.
* Informe de mejora. **2 SP**. – Álvaro D.
* Plan de pilotaje. **7 SP**. – Álvaro D.
  + Actualización.
  + Aplicación Fase 1 – Paso 2: Reunión.
* Buscar medida de rendimiento del equipo. **1 SP**. – Álvaro D.
  + Nº de SP sacados en cuanto tiempo.
* Video-Demo con Zoom. **1 SP.** – Jesús.
* Creación de batería inicial de ejercicios interactivos.
  + Ejercicio 1 (Diseño e Implantación). **3 SP.** – Álvaro D.
    - Corriente continua.
  + Ejercicio 2 (ídem) **3 SP.** – Jesús.
    - Estadística.
  + ~~Ejercicio 3 (ídem)~~ **~~3 SP.~~** ~~– Andrés.~~
    - ~~Análisis matemáticos.~~
  + ~~Ejercicio 4 (ídem)~~ **~~3 SP.~~** ~~– Andrés.~~
    - ~~Formulación inorgánica.~~
  + Ejercicio 5 (ídem) **3 SP.** – Ale G.
    - Geometría.

Diseño

* Página de bienvenida.
  + Contenido. **2 SP**. – Álvaro D.
  + Diseño. **3 SP**. – Ale G.
* Barra Navegación. **2 SP.** – Ale G.
  + Definir Navbar donde ir agregando la navegación.
* Estandarización. **3 SP**. – Ale G.
  + Formulario Tipo.
  + Listado Tipo.
  + Display Tipo.
* Diseño RESPONSIVE. **2 SP**. – Ale G.

Desarrollo de proyecto:

* ~~Automatización despliegue.~~ **~~6 SP.~~** ~~– Andrés.~~
* **Casos de Uso Usuario No Autenticado**
  + Registro como escuela completo. **7 SP**. – Álvaro S.
    - Gestión licencias específicas.
    - Paypal.
    - Diseño vista licencias.
* **Casos de Uso Programador**
  + Editar Perfil. **2 SP**. – Álvaro S.
    - Perfil Actor.
    - Contraseñas.
  + Listar los ejercicios del sistema. **2 SP**. – Jesús.
  + Listar ejercicios propios. **1 SP**. – Jesús.
  + Crear/modificar/eliminar ejercicios. Guardado inicial como draft hasta que el programador decida publicarlos. **20 SP**. – Migue y Ale R.

**AVANCE: 10 SP**

* + Ver su saldo. **2 SP**. - Jesús
* **Casos de Uso Escuela**
  + Editar Perfil. **1 SP**. – Jesús
    - Perfil Actor.
    - Contraseñas.
  + Listar los ejercicios del sistema. **1 SP**. – Jesús.
  + Listar ejercicios propios. **1 SP**. – Jesús.
  + Comprar ejercicios.
    - Elegir ejercicios gratis incluidos en la licencia.
    - Comprar real.
      * Cambios Modelo (Clase asociación). **1 SP.** – Álvaro S.
      * Paypal. **4 SP.** – Ale G.
      * Creación asignaturas al vuelo**. 7 SP**. – Álvaro S.
  + ~~Gestión de profesores.~~ **~~2 SP~~**~~. – Andrés.~~
    - ~~CRUD.~~
    - ~~Seguridad.~~
    - ~~Asignaciones con asignaturas.~~
  + Gestión de estudiantes. **2 SP**. – Jesús.
    - CRUD.
    - Seguridad.
    - Asignación con asignaturas.
* **Casos de Uso Profesor**
  + ~~Editar Perfil.~~ **~~1 SP~~**~~. – Andrés.~~
    - ~~Perfil Actor.~~
    - ~~Contraseñas.~~
  + Listar los ejercicios del sistema. **1 SP.** – Jesús.
  + Listar los ejercicios de su escuela. **1 SP.** – Jesús.
  + Listar los ejercicios que enseña. **1 SP.** – Jesús.
  + Visualizar el ejercicio y ejecutarlo. **6 SP. –** Ale R y Migue.

**AVANCE: 0 SP**

* **Casos de Uso Estudiante**
  + Editar Perfil. **1 SP**. – Jesús.
    - Perfil Actor.
    - Contraseñas.
  + ~~Listar las asignaturas en las que está matriculado.~~ **~~1 SP.~~** ~~– Andrés~~
  + Visualizar el ejercicio y ejecutarlo. **1 SP.** – Ale R y Migue

**AVANCE: 0 SP**

* **Testing:** Álvaro S., Jesús, Andrés, Ale G.
  + Documento en el que se vaya indicando lo que está implementado y casilla abierta para marcar como “Testeado”.
    - Desarrollador indica lo que ha implemntado y abre check. Indicar nombre del desarrollador.
    - Tester marca cuando lo compruebe:
      * Abre “Bug” en caso de que encuente errores y se lo dice al desarrollador.
      * Marca como Testeado y nombre del tester.
      * Tester debe contar las horas testeando.

**Próximos Sprint:**

* Filtrados
* Ordenaciones.
* Pago mensual a los programadores.
* Correcciones generales.