

## Desafío 1.

# ESCAPE WEB.

### ATAKEBUNE.

Un grupo de marines de la Marina Imperial Japonesa, se vio, no sabe muy bien cómo, encerrado en la bodega de un buque de guerra japonés de la era Sengoku (s. XVI).



Uno de los marines se da cuenta de que sus pies cada vez están más cubiertos de agua.

El objetivo es salir del barco en el menor tiempo posible, ya que calculan que en menos de media hora su vida corre peligro.

Para salir del barco, deberán encontrar las cinco llaves que les permitirán salir de la bodega... pero... ¿cómo?

### Objetivos

Vamos a realizar una página web que simule este **escape web**. En líneas generales, en esta página se podrán registrar usuarios que podrán:

- **Solicitar jugar una partida:** en este caso, se iniciará una partida cuando un mínimo de 3 jugadores y un máximo de 5 lo hayan solicitado.
- En una partida, el sistema irá lanzando a los jugadores de forma aleatoria enigmas. Si los jugadores los resuelven irán sumando llaves lo que irá puntuando de forma correspondiente. El primer marine que consiga una llave, y sólo éste, ascenderá a **almirante**. Un **almirante** podrá decidir qué jugadores resuelven los enigmas



incluido él mismo y a partir de ese momento no lo decidirá el sistema, a no ser que tarde demasiado. Cuando un marine falle un número de enigmas, el almirante puede decidir que pierden tiempo y no permitirle resolver enigmas y por tanto no puntuar.

- Saldrán del barco una vez hayan conseguido las 5 llaves necesarias.
- El sistema registrará para cada partida, los jugadores y roles que desempeñaron, el número de puntos, las llaves que consiguieron y el tiempo que tardaron en salir, si llegaron a conseguirlo.
- Además, también se podrá ver un ranking de jugadores.
- Cada jugador podrá ver las partidas que ha jugado, con quién...
- También podrá enviar mensajes a otros usuarios.

---

## Especificaciones

La aplicación constará de:

- Una página de inicio que permitirá entrar en el sistema. Esta página de inicio permitirá también, enlazar a una página de registro y a una página para restaurar contraseña.
- En la página de inicio tendremos una integración con reCaptcha de Google en el login.
- La página de registro permitirá dar los datos personales (incluyendo el email que servirá como usuario de inicio de sesión).
- Una vez dentro del sistema, como **usuario estándar**, podremos:
  - **Solicitar comenzar una partida.**
  - **Resolver enigmas dentro de una partida cuando le sean asignados.**
  - **Ver ranking de jugadores.**
  - **Ver las partidas que ha jugado y los datos de la partida.**
  - **Enviar mensajes a otros usuarios y ver los mensajes que le han sido enviados. A estos mensajes se les podrán adjuntar archivos, al menos uno:**



**como optativo, pero muy valorado, es que se puedan permitir adjuntar más de un archivo.**

- **Ver jugadores conectados en ese momento.**
- Como **usuario administrador**, podremos:
  - Con un CRUD, gestionar a los usuarios, altas, bajas, modificaciones, activaciones... los usuarios estarán desactivados por defecto hasta que los active el administrador. También se podrán crear otros administradores (o quitarles los privilegios a los actuales)
- Como **usuario editor**, podremos:
  - Gestionar las preguntas que el sistema lanza como enigmas.
  - Acceso a estadísticas internas sobre las partidas, preguntas...

Para cada módulo se tendrá en cuenta:

### **Desarrollo Web en Entorno Cliente.**

Código JavaScript estructurado y coherente, separado correctamente del resto de codificación.

Utilización correcta de funciones y estructuras de almacenamiento JavaScript.

Análisis del problema utilizando POO.

Utilización correcta de la cadena de prototipos de JavaScript.

Integración correcta de reCaptcha de Google (v3).

En cuanto a la metodología empleada para su desarrollo, será una metodología Scrum (ver anexo al final) y se valorará:

- Planificación de tareas por unidad funcional y coherencia.
- Sprints desarrollando lo planificado.
- Proyecto sincronizado con un repositorio en Github que se envía al profesor para su seguimiento.



### **Desarrollo Web en Entorno Servidor.**

Código PHP estructurado y ajustado al MVC.

Conexión con BD encapsulada y con POO.

Prevención de inyección de SQL.

Restauración de contraseña y confirmación de registro vía email.

Navegación coherente, uso correcto de las sesiones y código robusto con control de errores.

Todo lo recogido en el apartado “Desarrollo Web en Entorno Cliente” respecto a la metodología Scrum.

### **Diseño de Interfaces Web.**

Se crearan 2 diseños distintos (con sus correspondientes variantes en los diferentes tamaños ) y para cada uno se tendrá en cuenta lo siguiente:

- Estudio del usuario objetivo.
- Caminos de navegación por los wireframes
- Guía de estilo: colores primarios y colores secundarios, tipografía de títulos y párrafos, logo, iconos....
- El código debe estar estructurado en diferentes partials de SASS, con los oportunos comentarios del código. Aquí se valorará: uso adecuado de SASS. Complejidad VS necesidad, uso de CSS adecuado, confección de las reglas de forma optima, no forzar la inclusión de CSS, pero si intentar crear un diseño mas complejo.
- En cuanto a diseño se valorará: equilibrio de colores, colocación de elementos, adecuación de la estética al usuario objetivo.
- Se debe intentar usar diferentes elementos web para poder aplicar las reglas CSS.
- Estructura del HTML
- Independencia de la visualización respecto del código HTML. Se debe poder cambiar toda la visualización, simplemente cambiando la hoja de estilo. Los estilos deben diferenciarse no solo en la decoración CSS, sino en la ubicación y tamaño de los elementos.



**Despliegue de Aplicaciones Web.**

- Usar GitHub para el control de versiones permitiendo el acceso al repo a los profesores implicados.
- Se valorará un uso conveniente de la ramificación, para ello se deberán seguir como mínimo los siguientes requisitos.
  - ✓ Mantener la master para la versión estable del producto.
  - ✓ Crear una rama de desarrollo por unidad funcional.
- Se valorará el seguimiento de un flujo de trabajo como GitFlow.



## **ANEXO: METODOLOGÍA SCRUM**

SCRUM es una metodología ágil en la que se aplican regularmente un conjunto de buenas prácticas para trabajar en equipo y obtener el mejor resultado en el proyecto.

En SCRUM se realizan entregas parciales y regulares del producto final, priorizándolas según el beneficio que aportan al receptor del proyecto (cliente).

Los puntos destacables de esta metodología de trabajo son:

### **ROLES:**

- **Product Owner:** se encarga de analizar qué es lo que quiere el cliente final en su proyecto y lo detalla en una lista priorizada (**product backlog**).
- **Scrum Master:** organiza y facilita el trabajo. Defiende la metodología a llevar a cabo ante el product owner.
- **Team:** equipo de desarrolladores. Analizan las tareas del **product backlog**, detallan los ítems que van a contener, su prioridad y el tiempo que les va a llevar realizarlas.

**Una persona puede tener más de un rol. Por ejemplo, el scrum master puede formar parte del equipo de desarrollo.**

### **FUNCIONAMIENTO:**

El trabajo se organiza en **sprints**. Un sprint tiene un tiempo acotado (por ejemplo 2 semanas) y en él se incluyen las tareas (historias) establecidas en el **product backlog**.

El sprint comienza con una planificación (**planning**): se reúne el **team, product owner y scrum master**. Se analiza cada tarea establecida por el product owner en el backlog. Se prioriza, determina la dificultad técnica por el equipo de desarrollo y se establece el tiempo que tardará en completarse. Una vez analizada, se decide si esa tarea "**cabe**" en dicho sprint, en cuyo caso, el team se **COMPROMETE** a realizarla en dicho sprint.

Cuando finaliza el sprint, se realiza una **retrospectiva** analizando lo que se ha hecho bien y lo que se ha hecho mal, para proponer mejoras al respecto. Después se realiza una **demo** de las historias del sprint que se han **completado**, ante el **product owner** (y a veces ante el cliente final).

Cada día del sprint se hará una pequeña reunión denominada **daily** (tiene que ser corta (10') y se obliga a estar de pie para que no se alargue), en la que cada miembro del equipo de desarrollo cuenta en lo que ha trabajado, las dificultades y si necesita ayuda por parte del resto del equipo.

Cuando comienza un sprint, existe un tablero donde se encuentran todas las tareas sin asignar pendientes del sprint (**backlog**), y que cada desarrollador irá escogiendo (teniendo en cuenta al equipo) y asignando según sus preferencias, ritmo de trabajo, etc.

