Contenido

[1. Autores del trabajo, planificación y entrega 3](#_Toc510960280)

[1.1 Autores 3](#_Toc510960281)

[1.2 Planificación 3](#_Toc510960282)

[1.3 Entrega 4](#_Toc510960283)

[2. Descripción de las tecnologías 5](#_Toc510960284)

[2.1 Descripción de CraftyJS 5](#_Toc510960285)

[2.2 Descripción de la tecnología Phaser 7](#_Toc510960286)

[3. Criterios de comparación 9](#_Toc510960287)

[3.1 Categoría A: Generales 9](#_Toc510960288)

[**3.1.1 Criterio A.1: Desarrollador** 9](#_Toc510960289)

[**3.1.5 Criterio A.5: Popularidad** 9](#_Toc510960290)

[3.2 Categoría B: Tecnologías usadas 10](#_Toc510960291)

[**3.2.1 Criterio B.1: WebGL** 10](#_Toc510960292)

[**3.3.2 Criterio B.2: Arcade Physics** 10](#_Toc510960293)

[**3.3.3 Criterio B.3: Audio Web** 10](#_Toc510960294)

[3.3 Categoría C: Funcionalidad 11](#_Toc510960295)

[**3.3.1 Criterio C.1: Motor gráfico** 11](#_Toc510960296)

[**3.3.2 Criterio C.2: Api de Vibración** 11](#_Toc510960297)

[**3.3.3 Criterio C.3: Detección de colisiones** 11](#_Toc510960298)

[**3.3.5 Criterio C.5: Multiplataforma** 11](#_Toc510960299)

[**3.3.6 Criterio C.6: Partículas** 12](#_Toc510960300)

[4. Evaluación de los criterios por tecnología 13](#_Toc510960301)

[4.1 Evaluación de los criterios para la tecnología 1:CraftyJS 13](#_Toc510960302)

[4.2 Evaluación de los criterios para la tecnología 2: Phaser 14](#_Toc510960303)

[5. Comparación de las tecnologías 1](#_Toc510960304)

[6.1 Situación 1 1](#_Toc510960305)

[6.1.1 Descripción de la situación 1](#_Toc510960306)

[6.1.2 Recomendación de tecnología a utilizar 1](#_Toc510960307)

[6.2 Situación 2 1](#_Toc510960308)

[6.2.1 Descripción de la situación 1](#_Toc510960309)

[6.2.2 Recomendación de tecnología a utilizar 2](#_Toc510960310)

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

Jesús Tovar Armenteros.

Edgar Benítez Galán

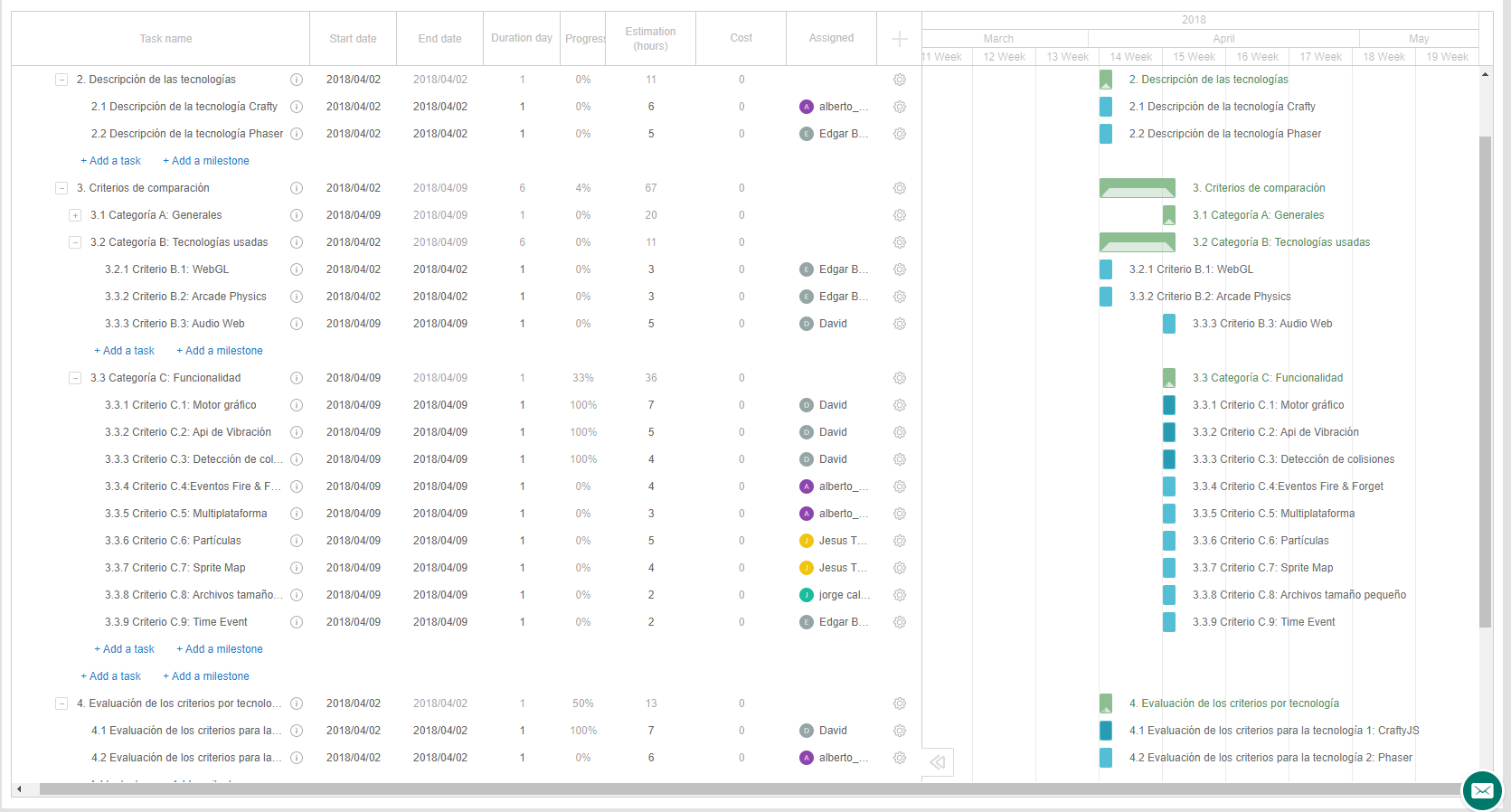
Alberto de Marcos Molina

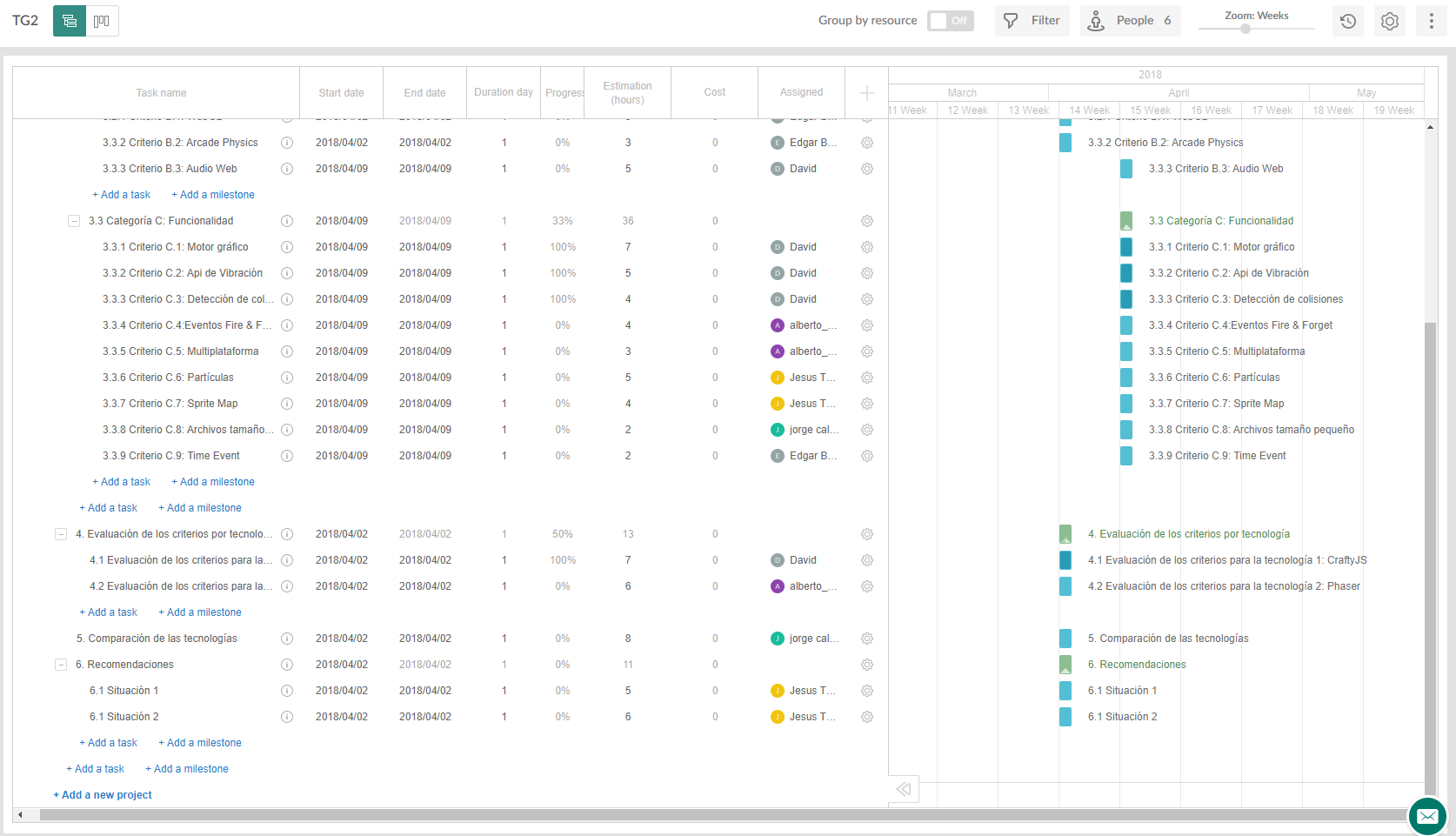
David García Cano

Jorge Calvo Carmona

## 1.2 Planificación

La planificación se ha realizado en la página web GanttPro. Hemos dividido los puntos de la siguiente forma:





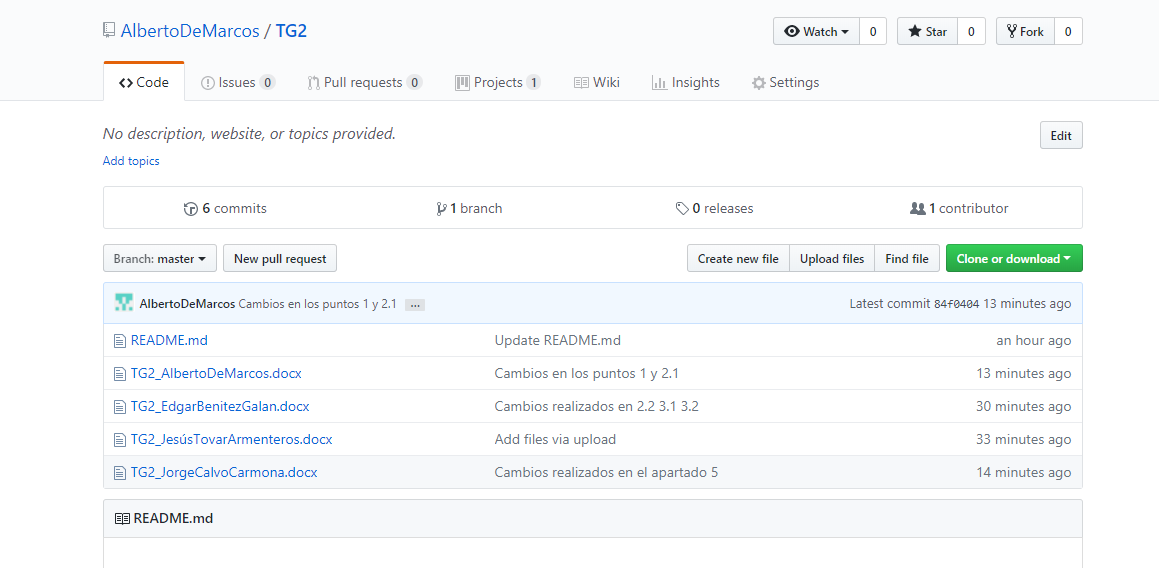
La URL donde se puede ver es la siguiente:

<https://app.ganttpro.com/#!/app/home>

## 1.3 Entrega

Hemos utilizado la plataforma Github para almacenar el TG2 (tanto Word como PowerPoint). La siguiente URL es del proyecto TG2:

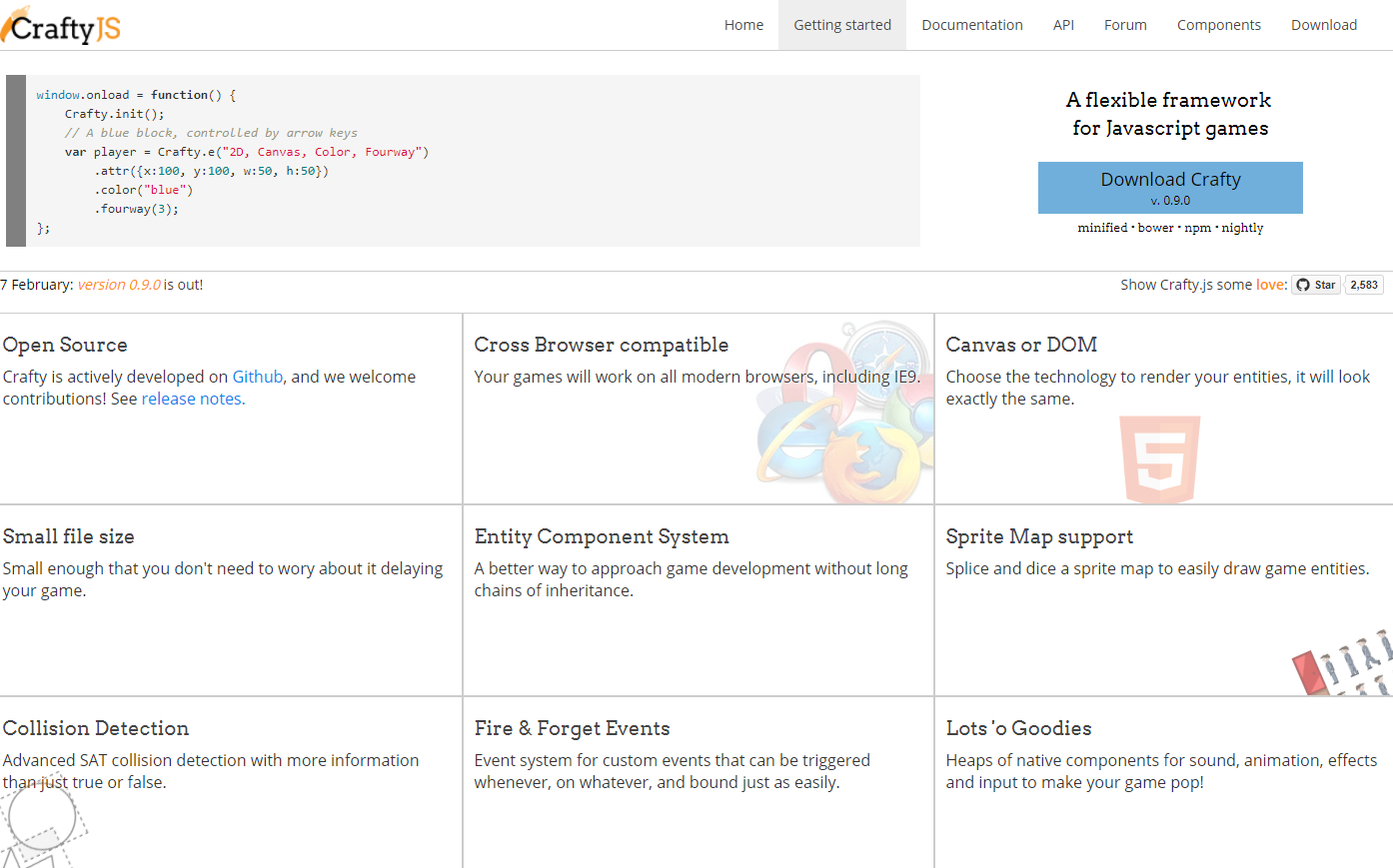
<https://github.com/AlbertoDeMarcos/TG2>



# 2. Descripción de las tecnologías

Hemos utilizado las siguientes tecnologías.

## 2.1 Descripción de CraftyJS

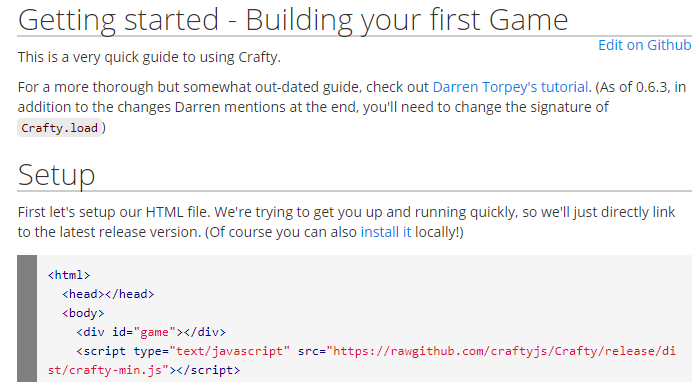
Fuente (Enlace):<http://craftyjs.com/>en este enlace podemos descargar el *Framework* además tiene enlaces para guías de cómo empezar a utilizarlo, una documentación, la API, descripción de componentes y foros.

Fuente (Enlace):<http://buildnewgames.com/introduction-to-crafty/>

CraftyJS es una colección cohesiva de bibliotecas Javascript creadas a medida que proporcionan un marco para la construcción basados en navegador que utilizan tecnologías web estándar. en otras palabras, es un motor de juego HTML5 basado en Javascript. está diseñado para que sea muy fácil hacer juegos simples con gráficos 2D.



Fuente (Enlace):<http://buildnewgames.com/introduction-to-crafty/>

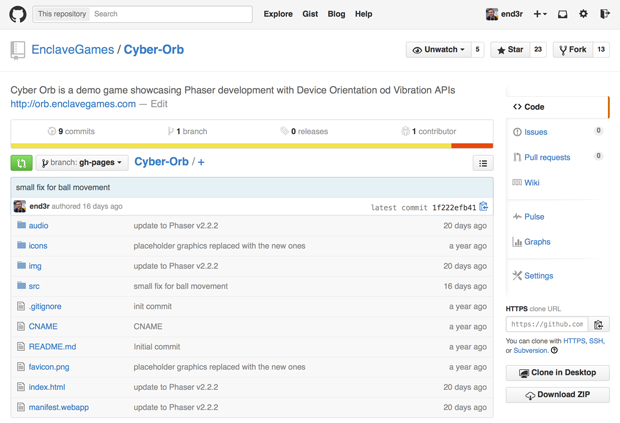
Instalación, primeros pasos: con este enlace tenemos una guía rápida que te explica paso a paso como comenzar el proyecto.

## 2.2 Descripción de la tecnología Phaser

Phaser es un *framework* para construir juegos, de móvil o escritorio, en HTML5. Es nuevo pero está creciendo velozmente gracias a la apasionada comunidad involucrada en el proceso de desarrollo. Puedes chequearlo en GitHub donde se encuentra como *open source.* Lee la documentación en línea y recorre su gran colección de ejemplos. El *framework* Phaser provee una serie de herramientas que acelerarán el desarrollo y te ayudarán a manejar las tareas genéricas necesarias para completar tu juego, para que así puedas concentrarte en la idea del juego en sí.

Los estados en Phaser son partes separadas de la lógica del juego, en nuestro caso los estamos cargando de archivos JavaScript independientes para un mejor mantenimiento. En este juego tenemos estados básicos: Boot, Preloader, MainMenu, Howto y Game. Bootse hará cargo de la inicialización de algunas opciones de configuración, Preloader cargará todos los elementos utilizados como los gráficos y el audio, MainMenu es el menú con el botón de inicio, Howto muestra las instrucciones de cómo jugar y el estado Game es el que finalmente te permite jugar el juego. Veamos rápidamente el contenido de esos estados.

La estructura de carpetas no es nada complicada: el punto de partida es el archivo index.html donde inicializamos el *framework* y configuraremos el canvas donde correrá el juego.



Puedes hacer clic en el archivo índex desde tu navegador favorito para iniciar el juego y probarlo. También hay tres carpetas en el directorio:

* img: Todas las imágenes que usaremos en el juego.
* src: Los archivos JavaScript que contienen todo el código fuente del juego definido dentro.
* audio: Los archivos de sonido usados en el juego.

URL:

<https://developer.mozilla.org/es/docs/Games/Workflows/HTML5_Gamedev_Phaser_Device_Orientation>

# 3. Criterios de comparación

## 3.1 Categoría A: Generales

### **3.1.1 Criterio A.1: Desarrollador**

Nombre del criterio: Desarrollador de la herramienta

Descripción: Nombre de la persona, institución o empresa que ha creado la herramienta.

Tipo de valor: Texto libre.

### **3.1.2 Criterio A.2: Madurez del Framework**

Nombre del criterio: Madurez.

Descripción: Indica si la madurez de este Framework tiene muchos años

Tipo de valor: Numérico.

### **3.1.3 Criterio A.3: Contiene la librería JavaScript**

Nombre del criterio: Contiene JavaScript.

Descripción: Indica si el framework contiene la librería JavaScript.

Tipo de valor: Booleano (Si/No).

### **3.1.4 Criterio A.4: Precio**

Nombre del criterio: Precio de la herramienta

Descripción: Si la herramienta es gratuita o de pago, y si es de pago saber cuánto cuesta.

Tipo de valor: Numérico

### **3.1.5 Criterio A.5: Popularidad**

Nombre del criterio: Popularidad de la herramienta.

Descripción: número y reputación de los usuarios, respaldo de la comunidad, grado de popularidad.

Tipo: texto libre.

**3.1.6 Criterio A.6: Soporte y documentación**

Nombre del criterio: Soporte y documentación

Descripción: Facilidad de un usuario para acceder a contenidos adicionales.

Tipo de valor: Texto libre.

**3.1.7 Criterio A.7: Usabilidad**

Nombre del criterio: Usabilidad

Descripción: Facilidad de uso de la herramienta

Tipo de valor: Texto Libre.

**3.1.8 Criterio A.8: Open Source**

Nombre del criterio: Open Source

Descripción: Herramienta de código abierto.

Tipo de valor: Boolean (Si/No).

**3.1.9 Criterio A.9: Instalación**

Nombre del criterio: Instalación

Descripción: Facilidad de instalación de la herramienta en los dispositivos.

Tipo de valor: Texto libre.

## 3.2 Categoría B: Tecnologías usadas

### **3.2.1 Criterio B.1: WebGL**

Nombre de criterio: WebGL

Descripción: define una API implementada en JavaScript para la renderización de gráficos en 3D dentro de cualquier navegador web

Tipo de valor: Boolean (Si/No).

### **3.3.2 Criterio B.2: Arcade Physics**

Nombre de criterio: Usa la librería arcade Physics

Descripción: Librería arcade Physics para dispositivos de todas las potencias

Tipo de valor: Boolean (Si/No).

### **3.3.3 Criterio B.3: Audio Web**

Nombre de criterio: Audio Web

Descripción: soporta Audio Web y directamente Audio HTML. Automáticamente controla el bloqueo en dispositivos, facilita la creación de audio para Sprites. Control de repeticiones y volumen.

Tipo de valor: Boolean (Si/No).

## 3.3 Categoría C: Funcionalidad

### **3.3.1 Criterio C.1: Motor gráfico**

Nombre del criterio: Motor gráfico

Descripción: uso del motor gráfico de renderizado 2D y 3D.

Tipo de valor: Boolean (Si/No).

### **3.3.2 Criterio C.2: Api de Vibración**

Nombre de criterio: Tiene funcionalidad de vibración

Descripción: efectos especiales con la ayuda de la API de Vibración para que el dispositivo vibre cuando el juego lo requiera

Tipo de valor: Boolean (Si/No).

### **3.3.3 Criterio C.3: Detección de colisiones**

Nombre de criterio: detección de colisiones

Descripción: detección de colisiones avanzado.

Tipo de valor: Boolean (Si/No).

**3.3.4 Criterio C.4:Eventos Fire & Forget**

Nombre de criterio: Eventos Fire & Forget

Descripción: Sistema de eventos personalizados que se pueden activar en cualquier momento y enlazados con facilidad.

Tipo de valor: Boolean (Si/No).

### **3.3.5 Criterio C.5: Multiplataforma**

Nombre de criterio: disponible para distintos dispositivos.

Descripción: posibilidad de utilizar la herramienta en diferentes sistemas operativos. Android,IOs...

Tipo de valor: Booleano (Si/No).

### **3.3**.6 Criterio C.6: Partículas

Nombre de criterio: Partículas

Descripción: Con su propio sistema de Partículas, permite crear divertidos efectos de partículas muy fácilmente. **Crea explosiones**, efectos de lluvia, fuego. O añade el emisor de partículas sobre un Sprite concreto.

Tipo de valor: Boolean (Si/No).

**3.3.7 Criterio C.7: Sprite Map**

Nombre de criterio: Sprite Map

Descripción: Empalma y recorta un mapa de sprites para dibujar fácilmente entidades de juego.

Tipo de valor: Boolean (Si/No).

**3.3.8 Criterio C.8: Archivos tamaño pequeño**

Nombre de criterio: Archivos tamaño pequeño

Descripción: Lo suficientemente pequeño para no preocuparte por el tiempo de carga en tu videojuego

Tipo de valor: Boolean (Si/No).

**3.3.9 Criterio C.9: Time Event**

Nombre del criterio: Time Event

Descripción: Posee un reloj interno propio para los eventos

Tipo de valor: Boolean (Si/No).

# 4. Evaluación de los criterios por tecnología

## 4.1 Evaluación de los criterios para la tecnología 1:CraftyJS

Debe incluir al menos una tabla con la siguiente estructura.

|  |  |
| --- | --- |
| Criterios | Evaluación |
| Criterio A.1: Desarrollador | Louis Stowasser |
| Criterio A.2: Madurez del framework | 8 años |
| Criterio A.3: Contiene la librería JavaScript | Si |
| Criterio A.4: Precio | Gratuito |
| Criterio A.5: Popularidad | Baja |
| Criterio A.6: Soporte y documentación | La página oficial de CraftyJS proporciona documentación en línea gratuita |
| Criterio A.7: Usabilidad | Permite el desarrollo de aplicaciones menos complejas |
| Criterio A.8: Open Source | Si |
| Criterio A.9: Instalación | Instalación sencilla |
| Criterio B.1: WebGL | Si |
| Criterio B.2: Arcade Physics | No |
| Criterio B.3: Audio Web | No |
| Criterio C.1: Motor gráfico | Si |
| Criterio C.2: Api de Vibración | No |
| Criterio C.3: Detección de colisiones | Si |
| Criterio C.4:Eventos Fire & Forget | Si |
| Criterio C.5: Multiplataforma | Android, IOS |
| Criterio C.6: Partículas | Si |
| Criterio C.7: Sprite Map | Si |
| Criterio C.8: Archivos tamaño pequeño | Si |
| Criterio C.9: Time Event | No |

## 4.2 Evaluación de los criterios para la tecnología 2: Phaser

|  |  |
| --- | --- |
| Criterios | Evaluación |
| Criterio A.1: Desarrollador | Empresa: Photon Storm |
| Criterio A.2: Madurez del framework | 2 años |
| Criterio A.3: Contiene la librería JavaScript | Si |
| Criterio A.4: Precio | Gratuito |
| Criterio A.5: Popularidad | Es nuevo pero está creciendo velozmente gracias a la apasionada comunidad involucrada en el proceso de desarrollo. |
| Criterio A.6: Soporte y documentación | La página oficial de phaser proporciona documentación en línea gratuita |
| Criterio A.7: Usabilidad | Permite fácil desarrollo gracias a su api simplificada |
| Criterio A.8: Open Source | Es una herramienta de código abierto que permite su modificación |
| Criterio A.9: Instalación | Fácil, disponible guía de ayuda online para la instalación. |
| Criterio B.1: WebGL | Si |
| Criterio B.2: Arcade Physics | *SI* |
| Criterio B.3: Audio Web | *Si* |
| Criterio C.1: Motor gráfico | Si |
| Criterio C.2: Api de Vibración | Si |
| Criterio C.3: Detección de colisiones | Si |
| Criterio C.4:Eventos Fire & Forget | No |
| Criterio C.5: Multiplataforma | Android, IOS |
| Criterio C.6: Partículas | Si |
| Criterio C.7: Sprite Map | No |
| Criterio C.8: Archivos tamaño pequeño | No |
| Criterio C.9: Time Event | Si |

# 5. Comparación de las tecnologías

|  |  |  |  |
| --- | --- | --- | --- |
| **Criterios** | **CraftyJS** | **Phaser** | **Comentarios** |
| **Generales** | | | |
| **Criterio A.1: Desarrollador** | Louis Stowasser | Empresa: Photon Storm | Desarrolladores |
| **Criterio A.2: Madurez del framework** | 8 años | 2 años | Crafty tiene mas años de desarrollo de framework y Phaser se encuentra en un crecimiento notable |
| **Criterio A.3: Contiene la librería JavaScript** | Si | Si | Ambos poseen la librería. |
| **Criterio A.4: Precio** | Gratuito | Gratuito | Los dos son gratuitos |
| **Criterio A.5: Popularidad** | Baja | Es nuevo pero está creciendo velozmente gracias a la apasionada comunidad involucrada en el proceso de desarrollo. | Crafty posee una baja popularidad mientras que Phaser está creciendo de forma rapida |
| **Criterio A.6: Soporte y documentación** | La página oficial de CraftyJS proporciona documentación en línea gratuita | La página oficial de phaser proporciona documentación en línea gratuita | Los dos poseen soporte y documentación es su página web |
| **Criterio A.7: Usabilidad** | Permite el desarrollo de aplicaciones menos complejas | Permite fácil desarrollo gracias a su api simplificada | Phaser permite un desarrollo mejor que Crafty debido a que te permite desarrollar tanto aplicaciones fáciles como complejas en un tiempo breve y mayor agilidad |
| **Criterio A.8: Open Source** | Si | Es una herramienta de código abierto que permite su modificación | Los dos son herramientas de libre acceso |
| **Criterio A.9: Instalación** | Instalación sencilla | Fácil, disponible guía de ayuda online para la instalación. | Ambos permiten una instalación sencilla |
| **Tecnologías usadas** | | | |
| **Criterio B.1: WebGL** | Si | Si | Permite usar la misma tecnología en sus SDK´s |
| **Criterio B.2: Arcade Physics** | No | *SI* | Phaser posee Arcada Physics para permitir acceso a dispostivos de cualquier potencia |
| **Criterio B.3: Audio Web** | No | *Si* | Phaser posee esta herramienta para la gestión de sonidos |
| **Funcionalidad** | | | |
| **Criterio C.1: Motor gráfico** | Si | Si | Ambos poseen motor grafico |
| **Criterio C.2: Api de Vibración** | No | Si | Phaser posee esta herramienta que permite la vibración |
| **Criterio C.3: Detección de colisiones** | Si | Si | Sirve para la detección de la colisión de objetos |
| **Criterio C.4:Eventos Fire & Forget** | Si | No | Crafty posee un sistema de eventos personalizados |
| **Criterio C.5: Multiplataforma** | Android, IOS | Android, IOS | Ambos se pueden usar en las mismas herramientas |
| **Criterio C.6: Partículas** | Si | Si | Ambos permiten crear efectos de particulas |
| **Criterio C.7: Sprite Map** | Si | No | Crafty permite cortar y empalmar un mapa de sprite |
| **Criterio C.8: Archivos tamaño pequeño** | Si | No | Crafty posee un sistema para que el usuario no se preocupe por la carga de su videojuego |
| **Criterio C.9: Time Event** | No | Si | Phaser posee un reloj interno para los eventos |

**6. Recomendaciones**

Deben plantearse posibles situaciones de uso, y recomendar justificadamente una u otra tecnología en función de la situación. Al menos 2 situaciones diferentes.

## 6.1 Situación 1

### **6.1.1 Descripción de la situación**

Una empresa quiere incluir un videojuego sencillo y que ocupe el menor espacio posible en su sistema de chat, en el cual prestan soporte técnico y asesoramiento a los clientes sobre problemas con los productos comprados o asesoramiento con sus compras. Debido a la demanda de la empresa el tiempo medio de atención es de 5 min por lo que quieren incluir un “pasatiempos” sencillo para amenizar la espera.

### **6.1.2 Recomendación de tecnología a utilizar**

En este caso el *framework* recomendado sería ***CraftyJS*** debido a que nos permite el desarrollo de un juego sencillo, en poco tiempo, compatible con la mayoría de los buscadores de internet, ya sea desde PC, móvil o tablet y además optimiza el espacio para que ocupe lo menos posible.

|  |  |  |
| --- | --- | --- |
| **Criterios relevantes para la decisión** | **Ventajas tecnología CraftyJS** | **Ventajas tecnología Phaser** |
| Small file size | Nos permite que nuestro juego ocupe el menor espacio posible | **-** |
| Sprite Map Support | Permite la importación de mapas ya creados con otras herramientas | **-** |
| Fire & Forget Events | Permite una fácil gestión de eventos, en este caso, el paso del juego durante la espera al chat cuando el técnico/asesor está disponible | **-** |

## 

## 6.2 Situación 2

### **6.2.1 Descripción de la situación**

En este caso nos encontramos con una empresa que quiere desarrollar un videojuego tipo arcade en el cual una nave espacial tiene que ir destruyendo meteoritos y otras naves enemigas, y debe incluir varios niveles, así como mejoras en la nave según se va avanzando a lo largo de la partida.

### **6.2.2 Recomendación de tecnología a utilizar**

En este caso el *framework* remendado sería ***Phaser***ya que posibilita en primer lugar la creación de videojuegos de mayor tamaño, en este caso es necesario, ya que se necesita incluir música, efectos de sonido, mejores gráficos, varias pantallas, efectos de explosiones y disparos, etc.

|  |  |  |
| --- | --- | --- |
| **Criterios relevantes para la decisión** | **Ventajas tecnología CraftyJS** | **Ventajas tecnología Phaser** |
| Audio Web | **-** | Lo que nos permite introducir una música de fondo, así como sonidos de disparos, colisiones, etc. |
| Gestión de tiempos | **-** | Posibilidad de incluir un reloj alternativo al proporcionado por el SO |
| API de vibración | **-** | Permite que la pantalla vibre cuando se produce una colisión |
| Posibilidad de creación de juegos de mayor tamaño | **-** | Da la posibilidad de crear videojuegos que ocupen mayor espacio que con Crafty lo que posibilita incluir pantalla, música y mejores gráficos. |