

**ESPECIFICACIÓN DE REQUISITOS.**

# **JÁBEGA**

## **TEAM**



# ÍNDICE

	<b>Páginas</b>
Datos de grupo:	1
Integrantes del grupo	1
Sección 1:	
Introducción	1
Sección 2:	
Requisitos funcionales	2-8
Requisitos no funcionales	8-10
Requisitos de dominio	10

# Integrantes

Nombre: Iván Díaz García, Correo: [ivandiazuni@gmail.com](mailto:ivandiazuni@gmail.com)

Nombre: Ángel Campos Salido, Correo: [angelcamsal06@uma.es](mailto:angelcamsal06@uma.es)

Nombre: Adrian Fernandez Vera, Correo: [adrianfeve@gmail.com](mailto:adrianfeve@gmail.com)

Nombre: Alberto Sánchez Aparicio, Correo: [albertoalual03@gmail.com](mailto:albertoalual03@gmail.com)

Nombre: Youcef Abi Ruiz, Correo: [youcefatbi@uma.es](mailto:youcefatbi@uma.es)

Nombre: Manuel Ruiz Campos, Correo: [manuelrc@uma.es](mailto:manuelrc@uma.es)

Nombre: Antonio Jesús Díaz Plaza, Correo: [antoniojesusdiazplaza@gmail.com](mailto:antoniojesusdiazplaza@gmail.com)

Nombre: Ángel Tobaruela Baños, Correo: [angeltoba@uma.es](mailto:angeltoba@uma.es)

## **Trello:**

<https://trello.com/invite/b/HS2u7KpC/ATTI56f83780148363e6d7423866ae20a00b3D028E4F/ingenieria-del-software-jabega-team>

## **Repositorio GitHub:**

<https://github.com/AngelUma/Jabega-Team>

# INTRODUCCIÓN- Sección 1

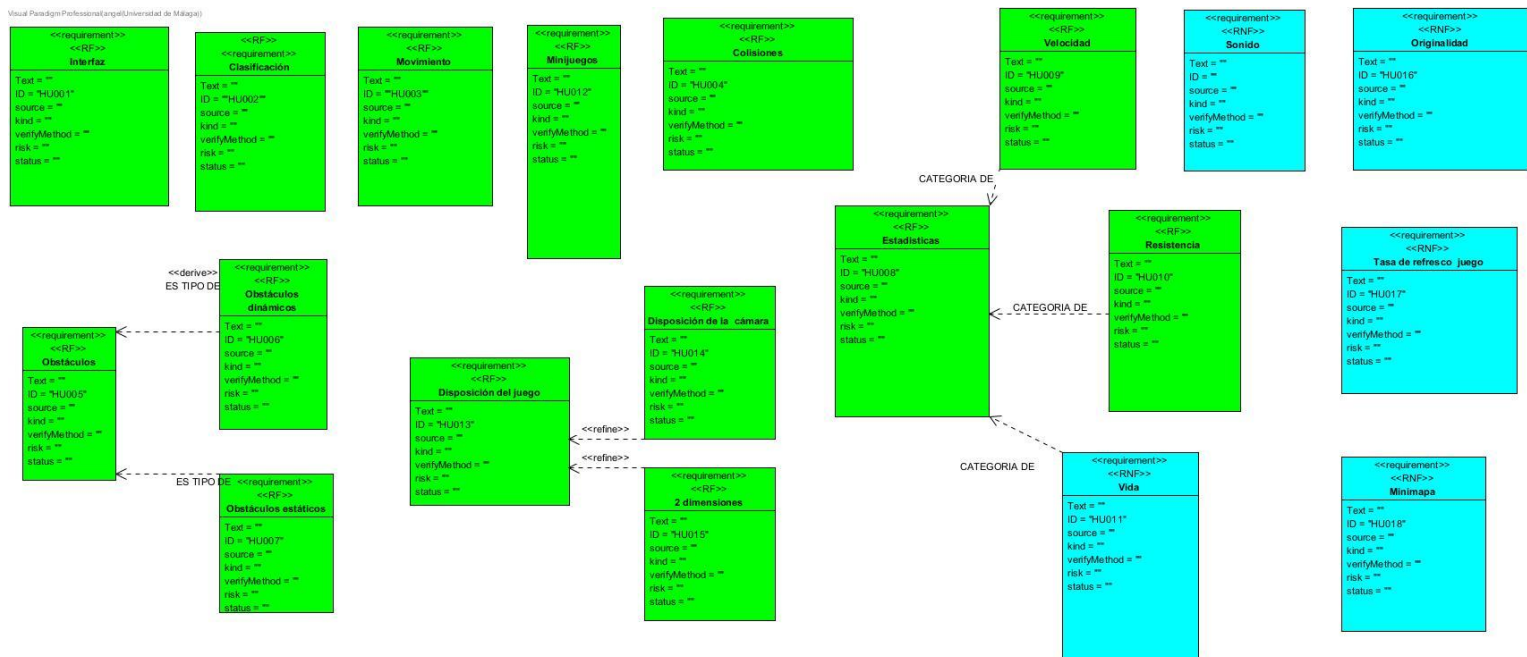
En este documento el equipo Jábega Team realizará una descripción sobre los requisitos necesarios para la realización del proyecto sobre la carrera de barcas.

Consistirá en una descripción detallada sobre los requisitos objetivos que cubre las necesidades del proyecto. Para su realización tuvimos una entrevista con el cliente, en la cual, realizamos diversas preguntas sobre los requisitos, tanto funcionales, como no funcionales, que nos ayudaron a hacernos una idea más clara del producto final que desea el cliente.

Para la composición de estos requisitos, hemos utilizado herramientas como Visual Paradigm Professional y el documento Word, que nos ayudarán a reflejar todos los requisitos junto a las relaciones que comparten entre ellos.

# REQUISITOS - Sección 2

Visual Paradigm Professional angul Universidad de Málaga



## Requisitos funcionales

Un requisito funcional es una declaración de cómo debe comportarse un sistema. Define lo que el sistema debe hacer para satisfacer las necesidades o expectativas del usuario. Los requisitos funcionales se pueden considerar como características que el usuario detecta. Son diferentes de los requisitos no funcionales, que definen cómo debe funcionar internamente el sistema (p. ej., rendimiento, seguridad, etc.).

Hacemos distinción entre cliente a la persona a la que se debe convencer de la validez de las suposiciones y decisiones del grupo y usuario al público que disfrutará del juego.

### HU001 Interfaz

El cliente quiere que el juego implemente un menú de inicio para empezar el juego, donde el usuario puede elegir fácilmente entre comenzar la partida, y algunas opciones más que son requisitos no funcionales. Tras finalizar la carrera dispondrá un posicionamiento para saber si ha conseguido pasar de ronda y acceder al minijuego y finalmente un podio con los resultados finales.

<<requirement>> <<RF>> <b>Interfaz</b>
Text = "" ID = "HU001" source = "" kind = "" verifyMethod = "" risk = "" status = ""

## HU002 Clasificación

El cliente nos ha pedido un juego clasificatorio que consta de una fase de clasificación y una fase final. Comienza la primera carrera, sí en ella el jugador ha conseguido ser primero o segundo pasará a la siguiente carrera. El objetivo es el de motivar al usuario a jugar nuestro hasta llegar a la final y conseguir el mejor resultado posible, la 1ª posición.

<<RF>> <<requirement>> <b>Clasificación</b>
Text = "" ID = ""HU002"" source = "" kind = "" verifyMethod = "" risk = "" status = ""

## HU003 Movimiento

Los barcos no se mueven por sí solos, el usuario debe apretar alguna tecla o no se moverán, no se podrán desplazar hacia atrás pero si pueden invadir otro carril. Chocar y girar o querer cambiarse de carril hará que su velocidad disminuya.

<<requirement>> <<RF>> <b>Movimiento</b>
Text = "" ID = ""HU003"" source = "" kind = "" verifyMethod = "" risk = "" status = ""

### HU004 Colisiones

Los barcos pueden colisionar con los obstáculos, con power ups, con los límites de su carril y con la meta, cada una de estas colisiones provocará distintas acciones, los power ups harán incrementar su velocidad o vida, los obstáculos y límites de carril la reducirán, y la meta provocará el final de la carrera.

<<requirement>> <<RF>> <b>Colisiones</b>
Text = "" ID = "HU004" source = "" kind = "" verifyMethod = "" risk = "" status = ""

### HU005 Obstáculos:

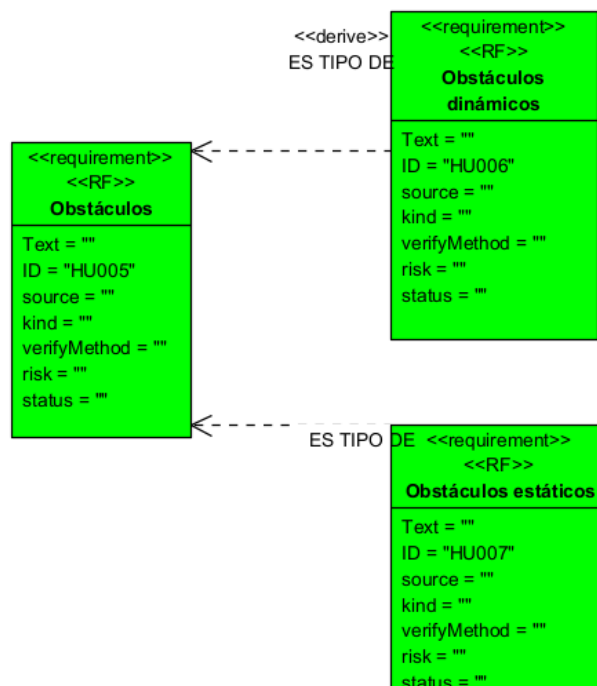
El cliente ha pedido que a lo largo del carril haya obstáculos con los que el usuario y la cpu interactúen con el objetivo de dificultar el juego para hacerlo más entretenido. Se distinguen dos variedades dentro de los obstáculos.

### HU006 Dinámicos:

A lo largo de la carrera encontraremos obstáculos que se estarán moviendo, encontraremos obstáculos con movimiento lateral en el carril, como patos y obstáculos con movimiento vertical que se moverán como si fueran arrastrados por la corriente como ramas.

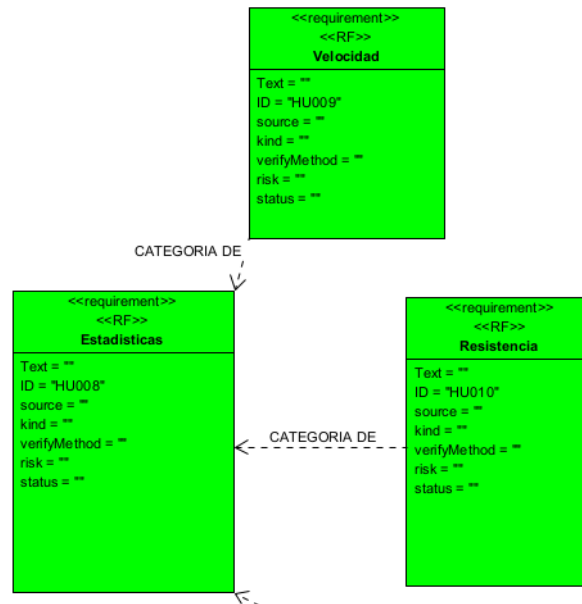
### HU007 Estáticos:

Durante una carrera a lo largo de cada carril encontraremos una serie de obstáculos que no se moverán a lo largo del carril.



## HU008 Estadísticas

El cliente ha pedido que las barcas cuenten con estadísticas que proporcionen variedad y jugabilidad al videojuego, dentro de cada categoría de estadística dispondremos de un rango del 1-5, siendo 5 la que cuente con esa mejor categoría de estadística. Las categorías son las siguientes:



## HU009 Velocidad:

Se basa en la velocidad fija en la que las barcas se desplazaran, sin contar su aceleración.

## HU010 Resistencia:

Cada barco tiene una resistencia, cuando un barco choca contra un obstáculo su resistencia irá disminuyendo, cuando dos barcos chocan puede que uno se rompa y otro no según sus resistencias(está relacionado con las colisiones ya que en las colisiones interviene la resistencia).



### **HU012 Minijuegos**

Entre las diferentes carreras se implementará un minijuego con relación al tema de los dragon boats, el cual tendrá afectación al juego. El tema queda a criterio del desarrollador. Se pretende que el juego no sea monótono para el usuario y esto sea una alternativa a la carrera principal.

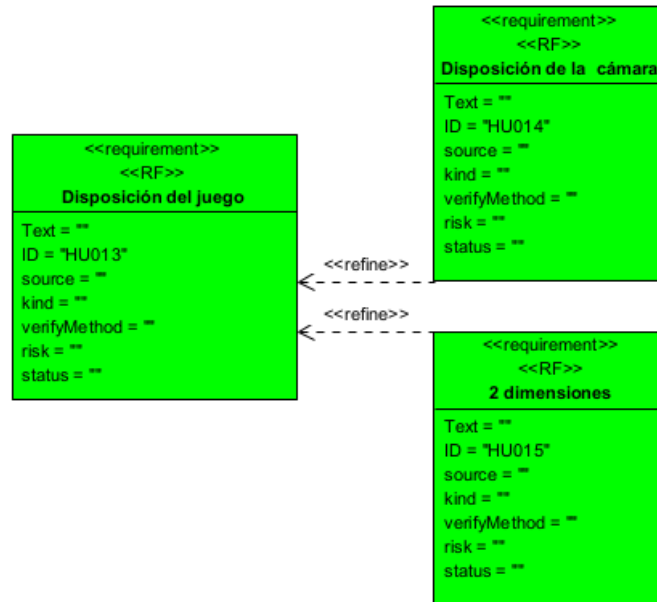
<b>&lt;&lt;requirement&gt;&gt;</b>
<b>&lt;&lt;RF&gt;&gt;</b>
<b>Minijuegos</b>
Text = ""
ID = "HU012"
source = ""
kind = ""
verifyMethod = ""
risk = ""
status = ""

### **HU013 Disposición del juego:**

La pantalla se dividirá en 4 carriles verticales que dividirán el río, también hay relación de dependencia con la disposición de la cámara.

### **HU014 Disposición de la cámara y HU015 2 Dimensiones:**

El cliente ha pedido que el juego sea mostrado de forma cenital, los ríos son vistos desde arriba. Los gráficos serán en 2D (una relación de dependencia con dimensiones, ya que si fuera en 3d por ejemplo no lo haríamos desde arriba).

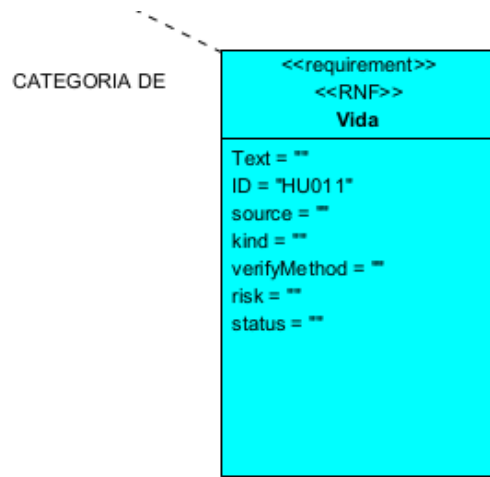


## Requisitos no funcionales

Los requisitos no funcionales son las restricciones o los requisitos impuestos al sistema. Especifican el atributo de calidad del software. Los requisitos no funcionales se ocupan de problemas como la escalabilidad, la mantenibilidad, el rendimiento, la portabilidad, la seguridad, la confiabilidad y muchos más. Los requisitos no funcionales abordan cuestiones vitales de calidad para los sistemas de software. En este artículo, intentaremos comprender los requisitos no funcionales en detalle con algunos ejemplos.

### HU011 Vida:

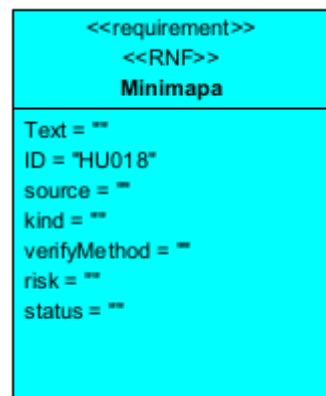
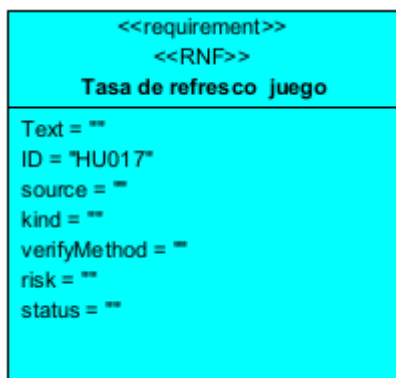
Es una estadística más que aporta variedad al juego. Cada barco tiene una barra de vida que va bajando cada vez que el barco se choca con algún obstáculo. Cuando esta barra llega a 0, el barco se hunde y pierdes la partida.



### HU016 Tasa de refresco(FPS) y HU017 Minimapa

El cliente prefiere una tasa de refresco de juego: La tasa de refresco del juego será preferiblemente de 30 FPS. El usuario tendría una mejor experiencia.

Como añadido el cliente ha sugerido tener un minimapa ubicado en una de las esquinas del escenario de la carrera con la posición simplificada de los 4 barcos referenciados con puntos móviles. Puede ser un gran atractivo para el usuario.



### UH Sonido

Para el usuario sería un gran atractivo que el juego disponga de una melodía interesante que lo haga más llamativo. Además el usuario podría elegir si prefiere jugar con sonido o sin sonido.

<<requirement>> <<RNF>> <b>Sonido</b>
Text = "" ID = "HU019" source = "" kind = "" verifyMethod = "" risk = "" status = ""

## Requisitos de Dominio

Los requerimientos del dominio se derivan del dominio de aplicación del sistema más que de las necesidades específicas de los usuarios. Normalmente incluyen terminología especializada del dominio o referencia a conceptos del dominio.

### HU018 Originalidad:

Es un requisito legal y de dominio, en el que se busca un juego innovador y original, para que como responsables no recibamos problemas legales sobre el plagio del software a otro proyecto.

<<requirement>> <<RNF>> <b>Originalidad</b>
Text = "" ID = "HU016" source = "" kind = "" verifyMethod = "" risk = "" status = ""