

Especificación de Requisitos según el estándar de IEEE 830

IEEE Std. 830-1998 22 de Octubre de 2008

Resumen

Este documento presenta, en español, el formato de Especificación de Requisitos Software (ERS) según la última versión del estándar IEEE 830. Según IEEE, este Documento de Requisitos, pese a no seguir obligatoria y estrictamente la organización y el formato dados en el estándar 830, sí incluye, de una forma o de otra, toda la información presentada en dicho estándar. El estándar de IEEE 830 no está libre de defectos ni de prejuicios, y por ello ha sido justamente criticado por múltiples autores y desde múltiples puntos de vista, llegándose a cuestionar incluso si es realmente un estándar en el sentido habitual que tiene el término en otras ingenierías. El presente documento no pretende pronunciarse ni a favor ni en contra de unos u otros: tan sólo reproduce, con propósitos fundamentalmente informativos, cómo se organizaría el proyecto de software a crear.

1. INTRODUCCIÓN

Este documento de especificación de requerimientos pretende mostrar lo que hará y lo que no hará el juego tipo “batalla naval” propuesto y llamado Battlecats, un juego que pretende dar una historia diferente a su tipo.

1.1 PROPÓSITO

Este documento va dirigido principalmente al Maestro Juan Carlos Perez Arriaga quien dirige este proyecto. Así mismo el equipo de desarrollo tomará de base este documento para el desarrollo e implementación del juego.

1.2 ÁMBITO DEL SISTEMA

Nombre del sistema: Battlecats Game

En la siguiente tabla se mostrará las diferentes funcionalidad en el sistema así como lo que no se hará y lo que se ha implementado en esta versión del juego.

Lo que el sistema hará	Lo que el sistema no hará
Se jugará en tiempo real	No se podrá jugar en un sistema diferente a Mac OS
Se jugará en computadoras diferentes	No se podrá jugar en redes diferentes
Se jugará a partir de una cuenta	No se reanudará partidas
Las partidas serán seguras	
Se podrá jugar mas de una partida a la vez	
Tendrá capacidad de internacionalización en Inglés y en español	

El juego servirá de entretenimiento y se jugará entre computadoras conectadas en la misma red, (Multijugador Local). El objetivo es mantener un juego estable para demostrar las habilidades en la Experiencia Educativa de “Tecnologías Para la Construcción de Software” Se espera que el sistema innove dentro de los alcances del juego de batalla naval.

1.3 Definición, Acrónimos y Abreviaturas

Caso de uso: es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso dentro del sistema.

CU: Caso de uso

Línea base: Es el punto de partida del sistema Battlecats Game

BG: Battlecats Game

Multijugador: Que se juega entre personas diferentes

Multijugador local: Que los jugadores están conectados a la misma red.

Internacionalización: La capacidad del sistema de reconocer el idioma en uso y aplicarlo al mismo.

1.4 Referencias

Por el momento este documento es independiente.

1.5 Visión general del documento

Este documento se estructurará tal como el estándar IEEE 830. Se explicará la línea base del juego Battlecats y explicará las funcionalidad paso por paso,

2. Descripción General

2.1 Reglas del juego .-

- Objetivo: Destruir la camada enemiga enemiga.
Barcos:
 - 5 gatos en un tablero de 25 posiciones.

En partida:

- ✖ Cada jugador coloca sus gatos en las casillas del tablero (Cada jugador tiene su propio tablero). El tablero de un jugador no puede ser visto por el otro jugador.
- ✖ Al inicio el jugador pulsa en las coordenadas a las que desea atacar(Un botón en un tablero dibujado por botones).
- ✖ Si en las coordenadas se encuentra un gato entonces el gato muere.
- ✖ Si no se encuentra ningún gato no pasara nada.
- ✖ Se deshabilitan las casillas en las que el jugador haya tirado para que él no repita posiciones en sus turnos.
- ✖ Cuando un jugador hace una jugada termina su turno
- ✖ Gana el jugador que destruya todos los gatos del otro jugador.

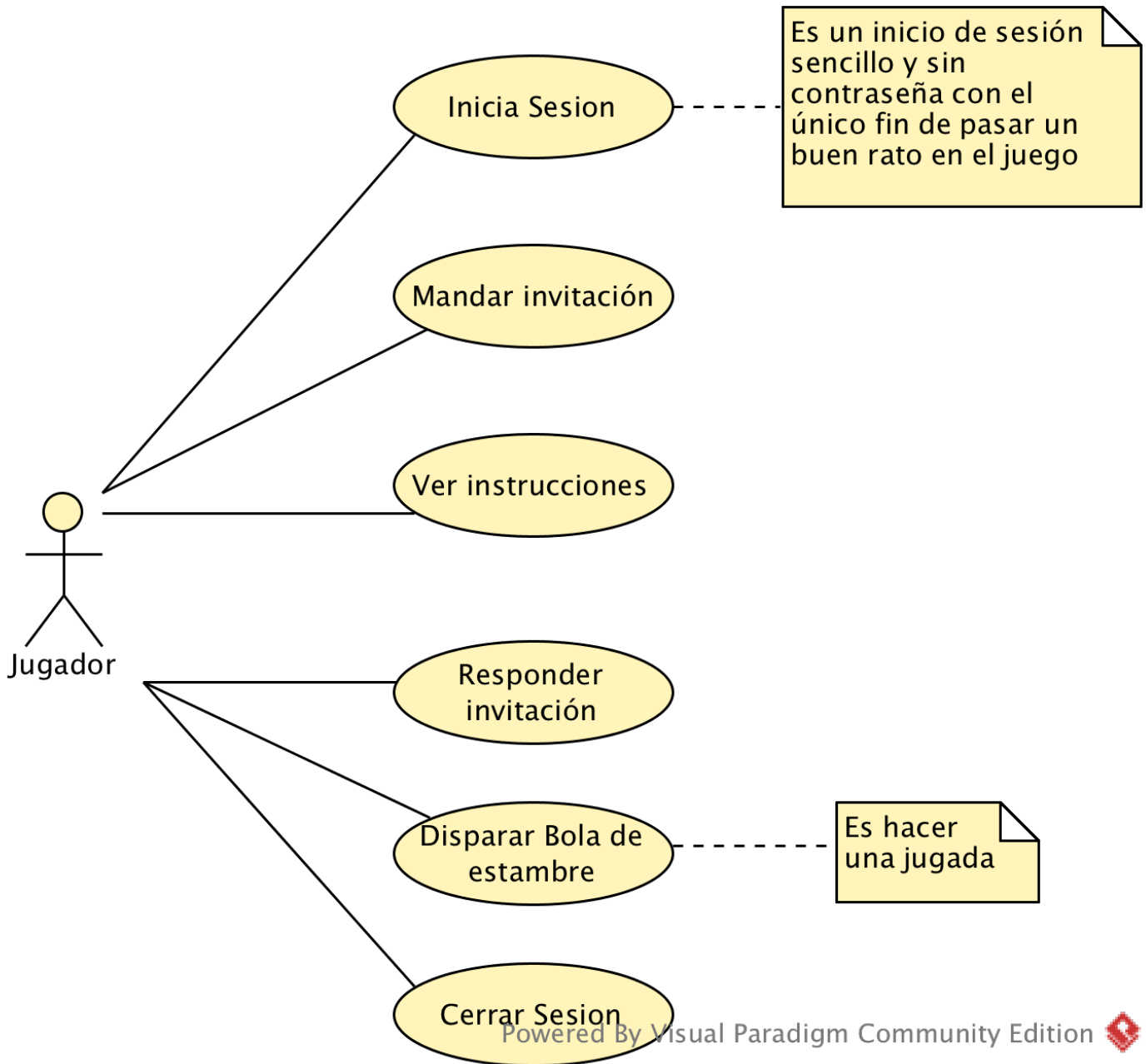
2.1 Perspectiva del producto

El producto es completamente independiente a otros sistemas.

2.2 Funciones del producto

A continuación se describen los casos de uso

Diagrama de casos de uso



3. Requerimientos no funcionales

Requerimientos de rendimiento.-

En cuanto al numero de jugadores que soporte el sistema guardados será de 10 usuarios conectados simultaneamente, tenga en cuenta que mientras más usuarios y partidas estén jugandose a la vez más lento responderá el servidor a peticiones.

Seguridad .-

Al usar swift la información se vuelve objetos lo cual podría funcionar como encriptación.

Fiabilidad.-

Siempre se podrá re-conectar un usuario, conservando su nombre sin que se vuelva a registrar en el sistema.

Disponibilidad.-

Aquí se pondrán los tiempos para las diferentes tipos de interacción con el sistema.

Mantenibilidad .-

En este caso se le dará mantenimiento cuando sea requerido por alguna falla, pero no se tiene contemplado que se le hagan revisiones por lapsos de tiempo.

Portabilidad.-

Solo podrá ser utilizado en sistemas que tengan OSX.

Caso de uso	Iniciar Sesión
Actores	Jugador
Descripción	El usuario se conectará con un nickname para jugar
pre-condición	Abrir el juego conectado al servidor
Flujo normal	<ol style="list-style-type: none"> 1. En la primera pantalla el Jugador selecciona "Empezar" 2. El sistema abre una ventana con un tablero y un campo de texto destinado para el nombre de usuario. 3. EL Jugador presiona 5 botones que representan las posiciones donde estarán sus gatos en el tablero. 4. El sistema al tener la quita posición abrirá una ventana de
Flujo alterno	-
Excepciones	<ol style="list-style-type: none"> 1. El jugador selecciona tablero sin poner nombre de usuario <ol style="list-style-type: none"> 1.1 El sistema muestra un mensaje: "Ingresa nombre de usuario" y regresa a la primera pantalla.
pos-condición	

Caso de uso	Disparar bola de estambre
Actores	Jugador
Descripción	El jugador selecciona una casilla del contrincante
pre-condición	Haber iniciado el juego
Flujo normal	<ol style="list-style-type: none"> 1. El jugador selecciona un gato en el mapa del atacante 2. El sistema modifica la imagen si acertó y bloquea la imagen para no permitir al usuario volver a tirar ahí
Flujo alterno	<ol style="list-style-type: none"> 1. El jugador no acierta el golpe 2. El sistema no cambia de imagen y solo bloquea el botón para no poder tirar de nuevo ahí
Excepciones	<ol style="list-style-type: none"> 1. Trata de tirar cuando no es su turno <ol style="list-style-type: none"> 1.1 El sistema en la parte inferior de la pantalla muestra un mensaje que le recuerda que no es su turno y debe esperar a que tire el otro jugador.
pos-condición	Haber generado un "disparo" según las reglas del juego (registrar el tiro)

Caso de uso	Mandar invitación
Actores	Jugador
Descripción	Se invita a otro jugador para empezar una partida
pre-condición	Haber iniciado sesión. Estar en la pantalla de los usuarios
Flujo normal	<ol style="list-style-type: none"> 1. El jugador pulsa dos veces en el nombre de otro usuario conectado (en verde) 2. El sistema manda una invitación al usuario a quien seleccionó 3. El usuario a quien se mandó la invitación acepta 4. El sistema abre la ventana del juego con los dos tableros.
Flujo alterno	2. El usuario invitado rechaza la invitación 2.1 Al jugador anfitrión le aparece una alerta: "Se ha rechazado tu invitación"
Excepciones	-
pos-condición	Se inicia una partida.

Caso de uso	Ver instrucciones
Actores	Jugador
Descripción	Ve las instrucciones de la aplicación
pre-condición	Haber iniciado sesión. Estar en la pantalla de los usuarios
Flujo normal	<ol style="list-style-type: none"> 1. El jugador selecciona el botón de ayuda ubicado en la esquina superior derecha 2. El sistema muestra una alerta con las instrucciones para invitar a alguien a jugar
Flujo alterno	-
Excepciones	-
pos-condición	-

Caso de uso	Responder invitación
Actores	Jugador
Descripción	Responde la invitación de un usuario
pre-condición	Haber iniciado sesión. Estar en la pantalla de los usuarios
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra una alerta que explica que un jugador lo ha invitado a jugar 2. El usuario acepta la invitación 3. El sistema abre la ventana del juego donde aparecen los tableros.
Flujo alterno	<ol style="list-style-type: none"> 2. El usuario rechaza la invitación 2.1 El sistema manda un mensaje al otro usuario que se ha rechazado la invitación 2.2 El sistema cierra la alerta pero del abierta la ventana de los usuarios
Excepciones	-
pos-condición	Empieza una partida.

Caso de uso	Responder invitación
Actores	Jugador
Descripción	Responde la invitación de un usuario
pre-condición	Haber iniciado sesión. Estar en la pantalla de los usuarios
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra una alerta que explica que un jugador lo ha invitado a jugar 2. El usuario acepta la invitación 3. El sistema abre la ventana del juego donde aparecen los tableros.
Flujo alterno	<ol style="list-style-type: none"> 2. El usuario rechaza la invitación 2.1 El sistema manda un mensaje al otro usuario que se ha rechazado la invitación 2.2 El sistema cierra la alerta pero del abierta la ventana de los usuarios
Excepciones	-
pos-condición	Empieza una partida.

Caso de uso	Cerrar Sesión
Actores	Jugador
Descripción	Se desconecta del juego
pre-condición	Haber iniciado sesión. Estar en la pantalla de los usuarios
Flujo normal	<ol style="list-style-type: none"> 1. El jugador selecciona la imagen del gato durmiendo 2. El sistema muestra un mensaje de comprobación si realmente desea salir 3. El jugador selecciona "Desconectar" 4. El sistema regresa a la primera ventana y desconecta al jugador
Flujo alterno	<ol style="list-style-type: none"> 3. El jugador selecciona "Cancelar" 3.1 El sistema cierra la alerta y permanece en la pantalla de usuarios
Excepciones	-
pos-condición	Desconecta al jugador

Apéndices.-

Restricciones del lenguaje

1. El sistema será programado en el lenguaje Swift 3, cuyas aplicaciones solo pueden ser ejecutadas en un entorno de iOS, apple-watch, apple-Tv y Mac OS como hemos mencionado en el apartado de portabilidad del tercer encabezado. Habiendo dicho, es imposible que se haga una versión para otros dispositivos y/o plataformas, puesto que ha sido acordado que será únicamente programado en este lenguaje.

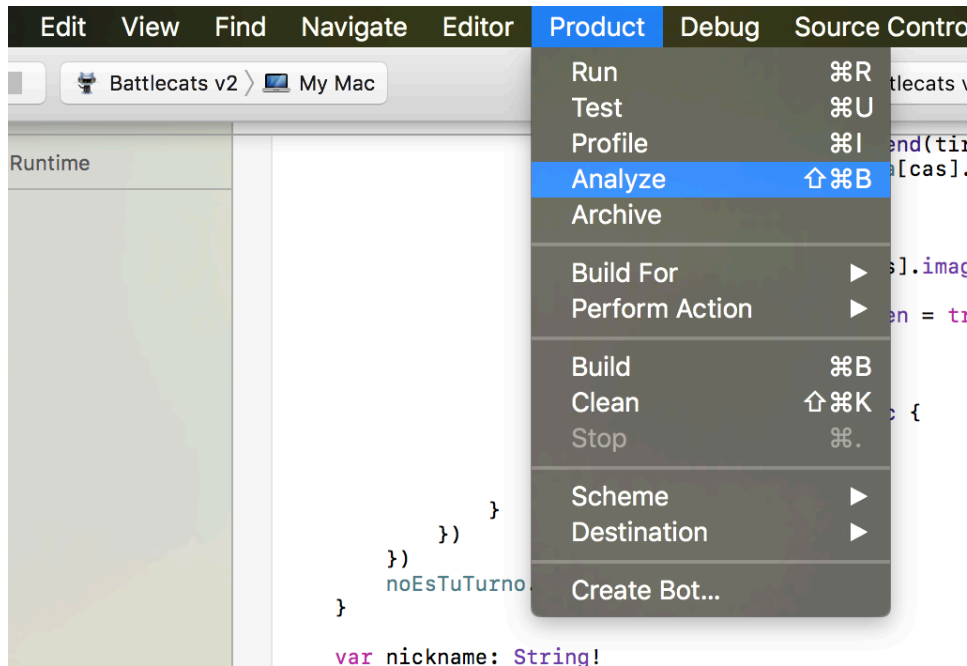
2. Se ha dispuesto el uso del framework Cocoa Application, (el nuevo framework surgido con Swift 3) este framework está dirigido a los sistemas operativos OSX lo cual significa que no soporta las características de iOS (como el tipo de ventanas y las características touch de los iPhone por ejemplo). Para jugarlo será necesaria una Mac o una Macbook.

Conclusión

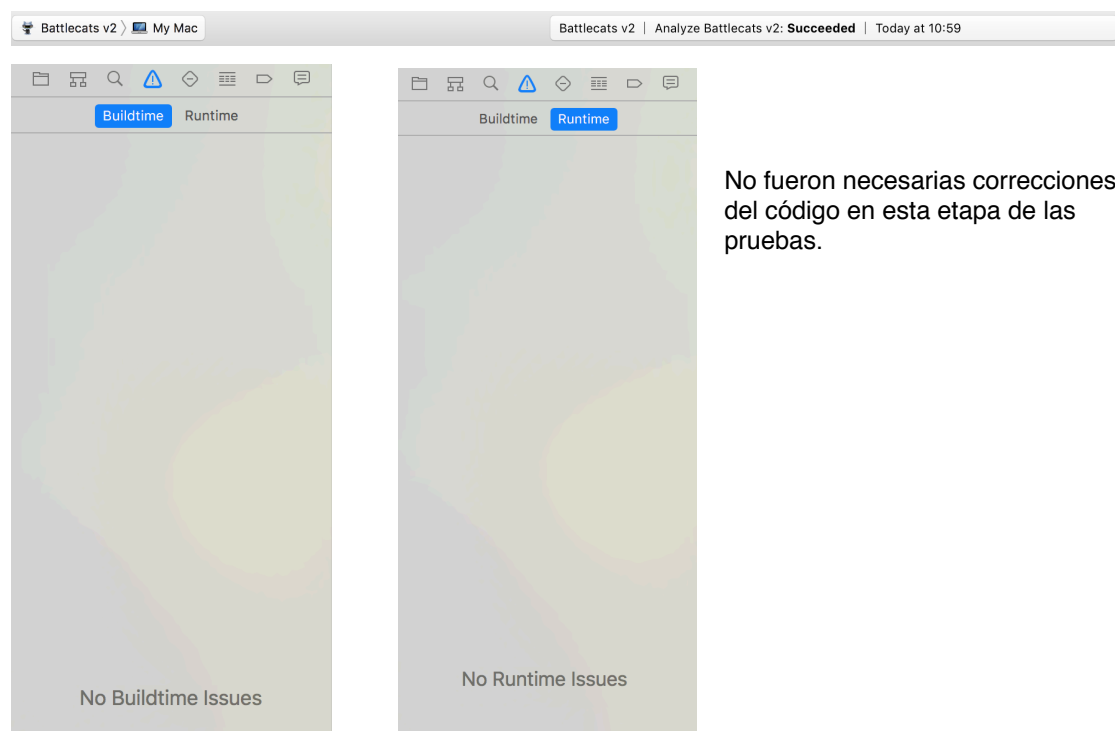
El uso de los frameworks que otorga Xcode, el IDE por excelencia del lenguaje Swift 3, facilita en gran manera la creación de una aplicación. Sin embargo las posibilidades que ofrece, al ser tantas, requiere de un riguroso estudio para poder aprovechar de manera óptima su potencial. El modelo y estilo de programación queda restringido al modelo-vista-controlador, que si bien es el modelo estándar para Swift y Objective-C no es el estilo acostumbrado de programación para la mayoría de los programadores, por lo que en el inicio fue complicado programar de esta manera.

Análisis de Código Estático

Gracias a la herramienta que proporciona Xcode, la cual analiza y da seguimiento a cada hilo de ejecución que se genera, y revisa a cada etiqueta, cuadro de texto y botón, tanto en funcionamiento como en la calidad de su título y traducción (si las hay); Se hizo el análisis de código estático, dando como resultado 0 Buldttime issues



Con el análisis exitoso:



No fueron necesarias correcciones del código en esta etapa de las pruebas.