

Planteamiento del problema:

En el desarrollo de campeonatos barriales de fútbol se puede observar la desorganización y pérdida de datos tanto de jugadores como de equipos ya que la información es manejada manualmente, lo que limita poder enviar información importante por parte de los gestores de los campeonatos hacia los participantes, dicha información se refiere a fechas de partidos, consulta de jugadores amonestados o cobro de multas por razón de tarjetas amarillas o rojas, razones por las cuales en todo campeonato barrial siempre existen discusiones o problemas debido a la falta de comunicados

Check list de Análisis

Atributo de calidad a considerar	Pregunta
Independencia del diseño	- ¿ La lista de requisitos anticipa el diseño o incluye información de implementación?
Concisión	- ¿ Cada requisito es simple o por el contrario, podría dividirse en varios requisitos? - ¿Existe algún requisito que no parezca añadir ninguna información útil acerca del sistema a desarrollar?
Realizabilidad	- ¿Es realizable el requisito en la plataforma de implementación? - ¿Existe algún requisito irrealizable con la tecnología actual? NOTA: Para responder a esta pregunta, es necesario conocer los aspectos técnicos de la plataforma donde se implementará el sistema
Consistencia externa	- ¿Existe algún requisito que contradiga requisitos organizativos explícitamente formulados?
Ambigüedad	- ¿Es posible interpretar de varias formas el requisito?
Verificabilidad	- ¿Es posible implementar algún caso de prueba que permita establecer que el requisito NO SE CUMPLE?

Ejemplos:

1. La lista de requisitos anticipa claramente información que se debe de tomar en cuenta para poder ejecutar y diseñar el programa de la vocalía de una liga barrial, ya que en esta pide que el programa permita registrar equipos de futbol de una liga barrial, datos del jugador, tarjetas rojas y amarillas, además que permita mostrar datos de los equipos, con el fin de que el programa lo puedan utilizar en la liga Barrial Bella Vista del Sur.
2. Los requisitos son simples y claros, además todos aportan información útil para la elaboración del software antes mencionado.
3. Teniendo la información y los datos de la liga barrial junto al entorno de desarrollo integrado de código abierto, y el lenguaje de programación java se puede realizar el programa con los paramentos que necesita el encargado de la Liga Barrial.
4. No existe un requisito que contradiga requisitos organizativos, al contrario todos muestran con claridad lo que se quiere lograr con el programa.
5. El requisito esta definido en un lugar y tiempo determinado lo que permite que el programa sea centrado y no se ingresen datos que no sean necesarios.
6. El sistema debe incluir a 12 equipos con un responsable, con sus respectivos jugadores, los cuales tienen nombre, edad, numero de camiseta, posición de la cancha. También el programa tiene que permitir el ingreso de tarjetas rojas y amarillas a los jugadores infractores, y las multas pertenecientes a esta falta.



Metodología de Desarrollo de Software – 7342

Cuestionario Grupal

Docente:

Ing. Jorge Raura

Integrantes:

- Marjorie Montalván
- Jessica Andrango
- Jordán Sánchez
- Kevin Topón

1. Consultar las metodologías más usadas actualmente en la industria de Software.

Características Metodologías	Objetivo	Función	Aspectos y/o fases
Extreme Programming XP	Ayudar en las relaciones entre los empleados y clientes	Potenciar las relaciones personales, a través, del trabajo en equipo, fomentando la comunicación y eliminando los tiempos muertos.	<ul style="list-style-type: none"> Planificación del proyecto con el cliente. Diseño del proyecto Codificación, donde los programadores trabajan en pareja para obtener resultados más eficientes y de calidad Pruebas para comprobar que funcionan los códigos que se van implementando
SCRUM	Abordar proyectos complejos que exigen una flexibilidad y una rapidez esencial a la hora de ejecutar los resultados.	La estrategia irá orientada a gestionar y normalizar los errores que se puedan producir en desarrollos demasiado largos, a través de, reuniones frecuentes para asegurar el cumplimiento de los objetivos establecidos.	<ul style="list-style-type: none"> Análisis. Desarrollo. Testing.
Kanban	Ayudar a mejorar la productividad y eficiencia del equipo de trabajo.	Elaborar un cuadro o diagrama en el que se reflejan tres columnas de tareas; pendientes, en proceso o terminadas	<ul style="list-style-type: none"> Planificación de tareas Mejora en el rendimiento de trabajo del equipo. Métricas visuales Los plazos de entregas son continuos.
Agile Inception	Orientada a los objetivos generales	Su meta es clarificar cuestiones como el tipo de cliente	<ul style="list-style-type: none"> Reuniones Intervenciones

	de una empresa independientemente.	objetivo, las propuestas de valor añadido, las formas de venta.	
Design Sprint, la metodología de Google	Mejorar los procesos que llevan a un producto o servicio mejorado y de calidad en el que los clientes cobran cada vez más importancia.	Realiza un proceso que dura 5 días en el que el negocio tiene que resolver todas las cuestiones relacionadas con diseño, prototipado, testeo de clientes.	La idea es que el trabajo se elabora en etapas de <i>sprints</i> en las que meses de trabajo se pueden reducir en pocas semanas

2. Definir el ciclo de vida de desarrollo aplicado a un caso práctico. Utilizar el proyecto planteado en la clase.

CASO PRÁCTICO

El director técnico de una liga barrial necesita una aplicación que ayude a mejorar y optimizar la información que genera antes y durante de cada partido, a su vez desea ingresar información de los equipos y los jugadores.



R1: El sistema debe permitir la administración de los equipos y jugadores (Ingreso, Modificación, eliminación) de equipos y jugadores.

R2: Un jugador puede contener la siguiente Información:

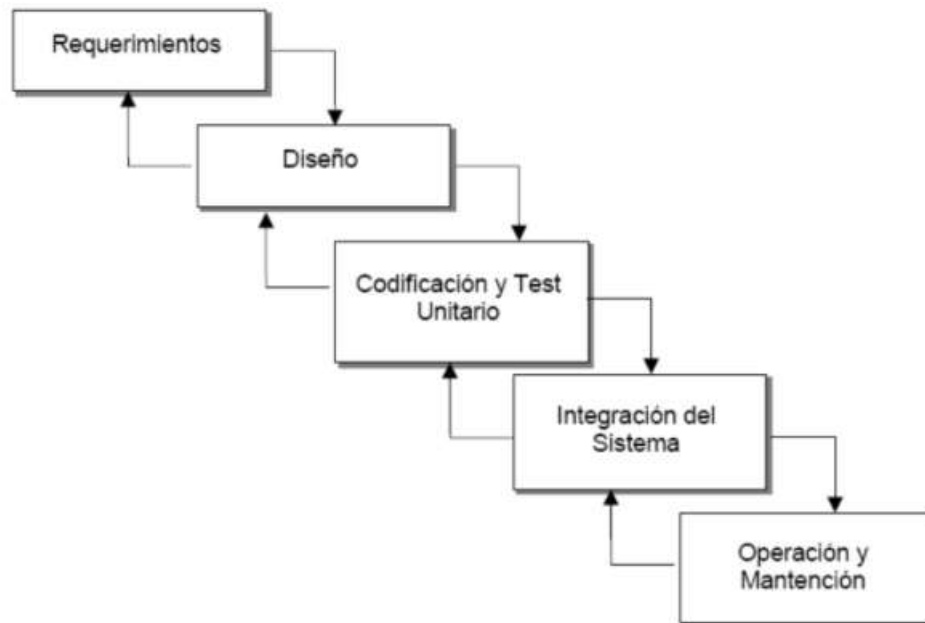
- Registro de nombre jugador
- Registro del nombre del equipo en el cual juega.
- Faltas
- Multas
- Agendar partidos

R3: El sistema permitirá consultar los datos del calendario por el nombre del equipo.

R4: El sistema generará un reporte del total faltas o multas por equipo y/o jugador.

R5: realizara las mismas funciones para cualquier escenario que se presente.

Nuestro modelo de ciclo de vida será en cascada debido a que conocemos todos los requerimientos que va a efectuar el programa.



Especificación de Requisitos según el estándar de IEEE 830

IEEE Std. 830-1998

22 de Octubre de 2008

Resumen

Este documento presenta, en castellano, el formato de Especificación de Requisitos Software (ERS) según la última versión del estándar IEEE 830. Según IEEE, un buen Documento de Requisitos, pese a no ser obligatorio que siga estrictamente la organización y el formato dados en el estándar 830, sí deberá incluir, 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 docentes, cómo se organizaría un Documento de Requisitos según el estándar IEEE 830.

Índice

1. Introducción	3
1.1. Propósito	3
1.2. Ámbito del Sistema	3
1.3. Definiciones, Acrónimos y Abreviaturas	3
1.4. Referencias.....	3
1.5. Visión General del Documento	4
2. Descripción General	4
2.1. Perspectiva del Producto	4
2.2. Funciones del Producto.....	4
2.3. Características de los Usuarios	5
2.4. Restricciones	5
2.5. Suposiciones y Dependencias	5
2.6. Requisitos Futuros.....	6
3. Requisitos Específicos	6
3.1. Interfaces Externas.....	7
3.2. Funciones	7
3.3. Requisitos de Rendimiento	9
3.4. Restricciones de Diseño	9
3.5. Atributos del Sistema	9
3.6. Otros Requisitos	9
4. Apéndices	9

1. Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el Sistema de base datos, la gestión de procesos y control de información para fomentar el deporte a través del orden . Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE 380.

1.1. Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para el desarrollo de un sistema de información que permitirá gestionar distintos procesos dentro de nuestra área de estudio (Liga Barrial). Éste será utilizado por personas que se encuentren interesadas en el deporte y sus directivos.

1.2 Ámbito del Sistema

- My Sport
- El programa puede registrar el nombre de los equipos, el nombre de los jugadores, además registrar faltas, multas y asignar fechas para próximos partidos.
- La comunidad y los directivos serán los beneficiados por nuestra aplicación , planteamos nuestro objetivo como el incentivar a los niños , jóvenes y adultos a realizar actividad física , atraves del orden optimizando el tiempo de las personas.

1.2. Definiciones, Acrónimos y Abreviaturas

Multas: Las sanciones disciplinarias son aquellas que afectan en particular al jugador que ha cometido la infracción.

Faltas/Sanciones: Existen dos tipos de sanciones disciplinarias: la amonestación (tarjeta amarilla) y la expulsión (tarjeta roja).

Calendarización: Fechas disponibles o no disponibles para agendar partidos.

1.3. Referencias

1.4. Visión General del Documento

Ampliar los recursos de nuestro programa, desarrollando un apoyo para otros deportes como básquet, boli y Natación ya que actualmente nos enfocamos directamente al futbol como actividad física debido a que es el deporte en el que participan e interactúan más jugadores.

2. Descripción General

- **Perspectiva del producto**

Generar una base de datos más extensa y fácil de utilizar con el fin de agilizar los procesos.

- **Funciones del producto**

Realiza la gestión de una base de datos, administrando y ordenando la información ingresada en el mismo, con el fin de evitar error o equivocaciones futuras.

- **Características de los usuarios**

Jugadores de futbol con conocimiento, nulo , medio ,básico o avanzado y que acepte al deporte como parte de su vida saludable.

- **Factores que se asumen**

Juego entre dos equipos conformados por 10 jugadores cada uno, en donde se van a especificar los motivos de las faltas y multas, como incumplimiento a las reglas planteadas en un reglamento previamente establecido.

- **Futuros requisitos**

Implementar nuevo deportes nuevas opciones de administración y mejor desempeño de nuestro programa en otros aspectos.

2.1. Características de los Usuarios

Para utilizar MySport se necesita:

- Niños mayores los 11 años, jóvenes y/o adultos.
- Conocimiento básico de computación
- Estudios medios (Básica y/o básicas superior)
- No necesita ningún tipo de experiencia.

MySport será un programa fácil de utilizar con una interfaz de usuario amigable.

2.2. Restricciones

- La falta de interés de las personas en realizar deporte, a muchas personas no les gusta el futbol , al igual que a muchas personas se niegan al uso de la tecnología como herramienta de apoyo.
- Adaptación a nuevos medios de administración.
- Criticalidad de la aplicación: Muchas personas consideran este programa como poco útil e innecesario.
- Aplicación funcional en distintos S.O.
- Acerca de la seguridad nuestra aplicación podría ser plagiada con gran facilidad.

2.3. Suposiciones y Dependencias

En el caso de que el S.O no sea compatible con la aplicación, además el hecho de que existan personas que consideren el programa con una complejidad alta de uso , confusiones de tipeo , un reseteo inesperado sin previa realización de copias de seguridad.

2.4 Requisitos futuros

- Implementar más funciones como un registro de jugadores habituales o poco habituales.
- Extender al registro de mas deportes ya que existen varias personas con interés nulo en el futbol
- Brindar una mejor interfaz de usuario para que nuestros clientes estén satisfechos.
- Modificar y extender su uso a distintas plataformas y software.

3. Requisitos Específicos

Se va a implementar un programa que va a ayudar a agilizar los procesos que se realizan dentro de un partido de futbol con el fin de fomentar el orden y disminuir la pérdida de tiempo que esto genera, de igual manera se pretende incentivar a las personas a realizar actividad física ya que con esta nueva aplicación los procesos serán más ágiles, sencillos y rápidos además de las distintas funciones que va a realizar nuestra aplicación como :

- Registro de Jugadores y equipos (Nombres).
- Registros de asistencia de jugadores y equipos.
- Registro de multas y faltas cometidas por los jugados y/o el árbitro.
- Calendarización para conocer cuáles son las fechas disponibles o no disponibles de ser el caso, además de podrá visualizar como se encuentran organizados os enfrentamientos.

3.1. Interfaces Externas

No se tiene claro si nuestro programa va a ser accesible or cualquier tipo de Software , dado el caso la interfaz no se mostrará de manera correcta.

3.2. Funciones

Esta subsección (quizá la más larga del documento) deberá especificar todas aquellas acciones (funciones) que deberá llevar a cabo el software. Nor-

malmente (aunque no siempre), son aquellas acciones expresables como “el sistema deberá ...”. Si se considera necesario, podrán utilizarse notaciones gráficas y tablas, pero siempre supeditadas al lenguaje natural, y no al revés. Es importante tener en cuenta que, en 1983, el Estándar de IEEE 830 establecía que las funciones deberían expresarse como una jerarquía funcional (en paralelo con los DFDs propuestos por el análisis estructurado). Pero el Estándar de IEEE 830, en sus últimas versiones, ya permite organizar esta subsección de múltiples formas, y sugiere, entre otras, las siguientes:

- Por tipos de usuario: Distintos usuarios poseen distintos requisitos. Para cada clase de usuario que exista en la organización, se especificarán los requisitos funcionales que le afecten o tengan mayor relación con sus tareas.
- Por objetos: Los objetos son entidades del mundo real que serán reflejadas en el sistema. Para cada objeto, se detallarán sus atributos y sus funciones. Los objetos pueden agruparse en clases. Esta organización de la ERS no quiere decir que el diseño del sistema siga el paradigma de Orientación a Objetos.
- Por objetivos: Un objetivo es un servicio que se desea que ofrezca el sistema y que requiere una determinada entrada para obtener su resultado. Para cada objetivo o subobjetivo que se persiga con el sistema, se detallarán las funciones que permitan llevarlo a cabo.
- Por estímulos: Se especificarán los posibles estímulos que recibe el sistema y las funciones relacionadas con dicho estímulo.
- Por jerarquía funcional: Si ninguna de las anteriores alternativas resulta de ayuda, la funcionalidad del sistema se especificará como una jerarquía de funciones que comparten entradas, salidas o datos internos. Se detallarán las funciones (entrada, proceso, salida) y las subfunciones del sistema. Esto no implica que el diseño del sistema deba realizarse según el paradigma de Diseño Estructurado.

Para organizar esta subsección de la ERS se elegirá alguna de las anteriores alternativas, o incluso alguna otra que se considere más conveniente. Deberá, eso sí, justificarse el porqué de tal elección.

3.3. Requisitos de Rendimiento

Se detallarán los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc.

También, si es necesario, se especificarán los requisitos de datos, es decir, aquellos requisitos que afecten a la información que se guardará en la base de datos. Por ejemplo, la frecuencia de uso, las capacidades de acceso y la cantidad de registros que se espera almacenar (decenas, cientos, miles o millones).

3.4. Restricciones de Diseño

Fácil de Utilizar

3.5. Atributos del Sistema

Se detallarán los atributos de calidad (las “ilities”) del sistema: Fiabilidad, mantenibilidad, portabilidad, y, muy importante, la seguridad. Deberá especificarse qué tipos de usuario están autorizados, o no, a realizar ciertas tareas, y cómo se implementarán los mecanismos de seguridad (por ejemplo, por medio de un *login* y una *password*).

3.6. Otros Requisitos

Cualquier otro requisito que no encaje en otra sección.

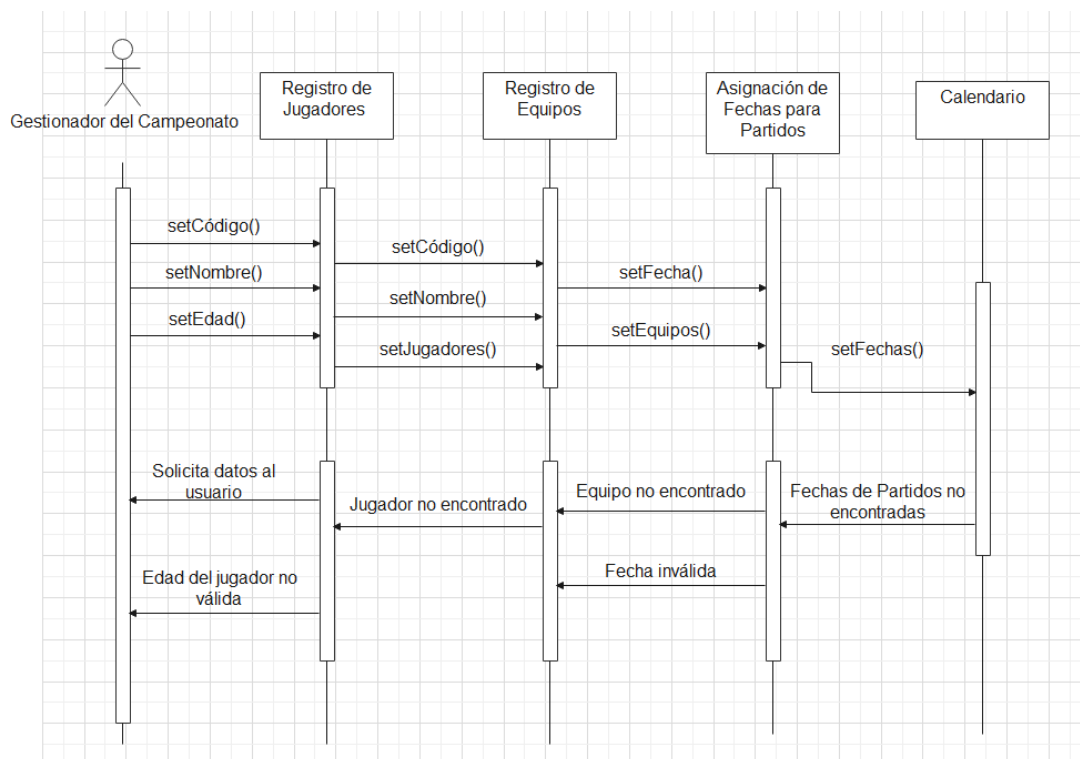
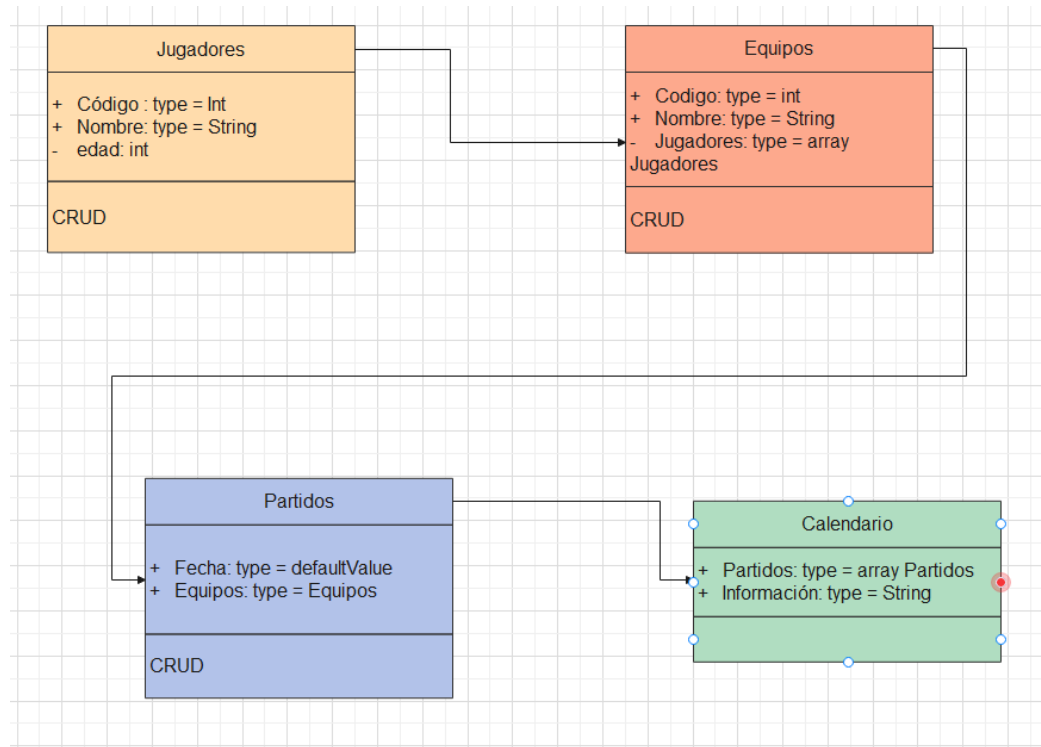
4. Apéndices

Pueden contener todo tipo de información relevante para la ERS pero que, propiamente, no forme parte de la ERS. Por ejemplo:

1. Formatos de entrada/salida de datos, por pantalla o en listados.
2. Resultados de análisis de costes.
3. Restricciones acerca del lenguaje de programación.

Cuestionario

Definir un diagrama de clases y diagramas de secuencia para un caso de estudio. Utilizar el proyecto planteado en clase



Definir casos de pruebas para una aplicación. Utilizar el proyecto planteado en clase.

Caja Negra

El software aplicado para la gestión de un campeonato de fútbol requiere de las siguientes entradas:

- Código de Equipo: Un valor de 3 dígitos, El primer dígito tiene que ser mayor a 1
- Nombre de Equipo: Una cadena de caracteres
- Edad de Jugadores: valor mayor a 16
- Código Jugador: Número de 10 dígitos, El primer dígito tiene que ser mayor a 1
- Nombre Jugador: Una cadena de caracteres
- Fecha partido: Una variable tipo date con formato dd//mm/aa
- Detalles del partido: Cadena de caracteres

Condición de Entrada	Tipo	Clase Equivalencia Válida	Clase Equivalencia No Válida
Código de Equipo	Valor	000<Código de equipo<= 999	Código de Equipo > 999 Código de Equipo < 001
Nombre de Equipo	Caracteres	Cualquier carácter es válido	No tiene restricción
Código Jugador	Valor	0000000000<Código de equipo<= 9999999999	Código de equipo > 9999999999 Código de equipo < 0000000001
Nombre Jugador	Caracteres	Cualquier carácter es válido	No tiene restricción
Edad de jugador	Valor	Edad de jugador >=16	Edad de jugador <16
Fecha partido	Date	dd/mm/aa	Todo tipo de dato diferente a date
Detalles del partido	Caracteres	Cualquier carácter es válido	No tiene restricción