

IEEE-830

16 de marzo de 2022

## Historial de cambios

crear tabla de el versionado de los cambios

# Índice

<b>1. Introducción</b>	<b>3</b>
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 . . . . .	3
<b>2. Descripción general del documento</b>	<b>3</b>
2.1. Perspectiva del producto . . . . .	3
2.2. Funciones del producto . . . . .	3
2.3. Características de los usuarios . . . . .	4
2.4. Restricciones . . . . .	4
2.5. Suposiciones y dependencias . . . . .	4
2.6. Requisitos futuros . . . . .	4
<b>3. Requisitos específicos</b>	<b>4</b>
3.1. Interfaces externas . . . . .	6
3.2. Funciones . . . . .	6
3.3. Requisitos de rendimiento . . . . .	6
3.4. Restricciones de diseño . . . . .	7
3.5. Atributos del sistema . . . . .	7
3.6. Otros requisitos . . . . .	7
<b>4. Apéndices</b>	<b>8</b>

# 1. Introducción

## 1.1. Propósito

Este documento presenta una especificación de requerimientos de software para un sistema de posicionamiento de antena. Este dispositivo generalmente se conoce como rotador.

Esta dirigido a técnicos, profesionales y operarios que intervengan en los sistemas de apuntamiento que posee el IAR.

## 1.2. Ámbito del sistema

Este sistema, se desarrolla como un subsistema del interferómetro MIA(<https://www.iar.unlp.edu.ar/sli>) y el proyecto de construcción de estaciones terrenas. Se utilizar para realizar el apuntamiento de radiofuentes, y el seguimiento de satélites en forma automática. El nombre del sistema rotador será ROT\_IAR. Adicionalmente, tiene expectativas de escalar, y realizar una producción en serie.

## 1.3. Definiciones, Acrónimos y Abreviaturas

ver los acrónimos al terminar el documento

## 1.4. Referencias

ver tesis, y plan de trabajo !

## 1.5. Visión general del documento

Este documento se realiza siguiendo el estándar IEEE Std. 830-1998

# 2. Descripción general del documento

## 2.1. Perspectiva del producto

El software es parte de un sistema mayor, denominado interferómetro MIA y estaciones terrenas. Este sistema de apuntamiento, se adicionará al sistema mecánico de la antena que está en fase de construcción. Este sistema, realizará el apuntamiento de antena, y este según el proyecto(MIA o estaciones terrenas) es automático o manual. El diagrama en bloques del sistema se muestra en la figura 1.

El presente documento describe los requerimientos de software del bloque single board computer, y las interfaces del sistema que se observa en la figura 1.

## 2.2. Funciones del producto

1. Servidor web embebido
2. Compatible con el software Gpredict y Stellarium, y scripts de antenas principales
3. Reinicio del en forma remota
4. Interrupción de operación en caso de condiciones climáticas adversas.
5. información de la operación y estado actual del sistema(tracking, untracked, y cenit).

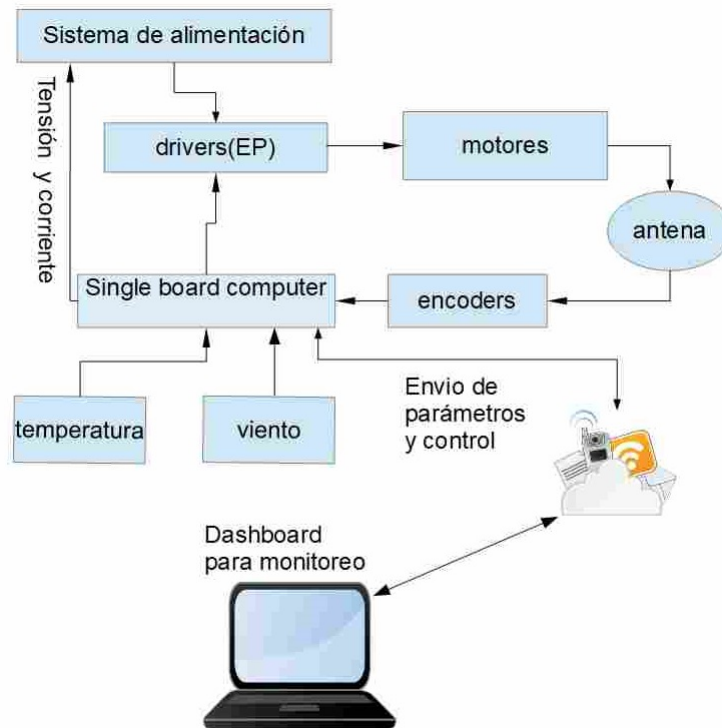


Figura 1: Diagrama en bloques del sistema

### 2.3. Características de los usuarios

Los usuarios serán técnicos, operarios y profesionales con conocimiento y experiencia en los sistemas de apuntamiento y manejo de rotadores.

### 2.4. Restricciones

- Lenguaje python3 por cuestiones de compatibilidad de scripts de manejo principal de las antenas.
- El software debe estar bajo control de versiones.

### 2.5. Suposiciones y dependencias

### 2.6. Requisitos futuros

Es deseable que el sistema posea control de velocidad.

## 3. Requisitos específicos

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la sección más larga e importante de la ERS. Deberán aplicarse los siguientes principios:

- El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.
- Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.
- Todo requisito deberá ser unívocamente identificable mediante algún código o sistema de numeración adecuado.
- Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características:
  - **Corrección:** La ERS es correcta si y sólo si todo requisito que figura aquí(y que será implementado en el sistema) refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el deseado.
  - **No ambiguos:** Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en glosario.
  - **Completos:** Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válidos como no válidos.
  - **Consistentes:** Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorios no es implementable.
  - **Clasificados:** Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esenciales, condicionales u opcionales) o por estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requisitos no esenciales.
  - **Verificables:** La ERS es verificable si y sólo si todos sus requisitos son verificables. Un requisito es verificable (testable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verificable. Expresiones como a veces, bien, adecuado, etc introducen ambigüedad en los requisitos. Requisitos como ".<sup>en</sup> caso de accidente la nube tóxica no se extenderá más allá de 25km" no es verificable por el alto costo que conlleva.
  - **Modificables:** La ERS es modificable si y sólo si se encuentra estructurada de forma que los cambios a los requisitos puedan realizarse de forma fácil, completa y consistente. La utilización de herramientas automáticas de gestión de requisito (por ejemplo RequisitePro o Doors) facilitan enormemente esta tarea.
  - **Trazables:** La ERS es trazable si se conoce el origen de cada requisito y facilita la referencia de cada requisito a los componentes y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc) de cada requisito. La trazabilidad hacia delante de un requisito R indica qué componentes del sistema son los que realizan el registro R.

### 3.1. Interfaces externas

Se describirán los requisitos que afecten a la interfaz de usuario, interfaz con otros sistemas (hardware y software) e interfaces de comunicaciones.

### 3.2. Funciones

Esta subsección (quizás la más larga del documento) deberá especificar todas aquellas acciones (funciones) que deberá llevar a cabo el software. Normalmente (aunque no siempre) son aquellas acciones expresables como ".<sup>el</sup> 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 propuestas 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 usuarios: 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 identidades 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 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

Todo aquello que restrinja las decisiones relativas al diseño de la aplicación: Restricciones de otros estándares, limitaciones del hardware, etc.

### 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 usuarios 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

Puede 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.