Monitor de Sistema

Documento de Visión

Preparado por Ortega y Páez

para <<Sistemas Operativos>>

Versión 1.0

Fecha: 24-09-2016

Historial de Revisiones.

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Autor | Descripción |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Aprobado por | Información de aprobación. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Historial de Aprobaciones.

[1. Introducción. 2](#_Toc243196482)

[1.1 Propósito. 2](#_Toc243196483)

[1.2 Postulado del problema. 2](#_Toc243196484)

[2. Oportunidades de negocio. 3](#_Toc243196485)

[3. Solución Propuesta. 4](#_Toc243196486)

[3.1 Requerimientos funcionales. 4](#_Toc243196487)

[3.2 Requerimientos no funcionales. 4](#_Toc243196488)

[4. Restricciones. 4](#_Toc243196489)

[5. Riesgos. 6](#_Toc243196490)

# 

# 1. Introducción.

## 1.1 Propósito.

El propósito del presente documento es el de registrar la visión general del Software a realizar por Ortega y Páez para la materia de Sistemas Operativos; así también como documentar los riesgos y restricciones del mismo.

## 1.2 Postulado del problema.

El Monitor de Sistema será responsable del manejo y monitoreo de los procesos que se encuentran listos, en ejecución y dormidos en el Sistema.

# 2. Oportunidades de negocio.

# 3. Solución Propuesta.

En esta sección se describen los requerimientos tanto funcionales como no funcionales del sistema.

## 3.1 Requerimientos funcionales.

* El Software debe mostrar al usuario los procesos que se encuentran en:
* Ejecución
* Listos
* Dormidos
* El Software debe permitir al usuario realizar operaciones con los procesos, tales como:
* Establecer prioridad
* Abrir ubicación del archivo
* Terminar proceso
* Mostrar el número de hilos que dicho proceso tiene
* Mostrar procesos hijos
* Mostrar procesos padres

## 3.2 Requerimientos no funcionales.

* El Software debe trabajar de forma concurrente
* El Software debe incluir una interfaz gráfica intuitiva y sencilla
* El Código fuente debe estar comentado correctamente
* El Código fuente debe estar bien estructurado

# 

# 4. Restricciones.

* El Software debe trabajar de forma concurrente
* El Software debe tener una historia en git no menor a 5 commits
* El Código fuente debe estar comentado correctamente
* El Software debe incluir una sección de presentación donde explique:
  + - * Problema
      * Lógica de Operación
      * Ejemplos de uso e invocación

# 5. Riesgos.

* xxx.
* xxx.
* xxx.