**INGENIERÍA DE SOFTWARE**

**RÚBRICA PARA EVALUAR El DOCUMENTO DE ARQUITECTURA DE SOFTWARE**

|  |  |
| --- | --- |
| **NOMBRE DEL PROYECTO :** |  |
| **EVALUADO POR :** |  |
| **FECHA DE EVALUACIÓN:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **C0. Rúbrica Cero[[1]](#footnote-1)** | | | **0** |
|  | **C0.1 Cumplimiento de la rúbrica Cero** | **Suficiente:** El documento cumple plenamente con la rúbrica Cero. | **0** |
|  |  | **Insuficiente:** El documento no cumple con la rúbrica Cero. | **-20** |

|  |  |  |  |
| --- | --- | --- | --- |
| **C1. Definición** | | | **20** |
|  | **C1.1** Vista lógica | **Insuficiente:** No describe todas las partes significantes del sistema, según los requisitos funcionales del mismo. | **0** |
| **** |
|  |  | **Necesita Mejorar:** Define todas las partes significantes del sistema. Basado en el dominio del problema, crea clases, realiza un diagrama de estados (de ser requerido), identifica clases pasivas, activas, protegidas, identifica persistencia y subordinación entre los objetos, pero no las agrupa de forma que no le permiten diseñar paquetes ni mucho menos encontrar mecanismos o elementos comunes de diseño en diversas partes del sistema. | **2** |
| **[4.0 / 20.0]** | **** |
|  |  | **Esperado:** Define todas las partes significantes del sistema. Basado en el dominio del problema, crea clases, realiza un diagrama de estados (de ser requerido), identifica clases pasivas, activas, protegidas, identifica persistencia y subordinación entre los objetos. Identifica los mecanismos y servicios comunes de las clases presentadas. Las clases presentadas son agrupadas y presentadas bajo un subsistema o paquete. | **4** |
|  | **** |
|  | **C1.2 Vista de proceso** | **Insuficiente:** No toma en cuenta los requisitos no funcionales del sistema, especialmente los relacionados a performance y disponibilidad. No identifica procesos en las actividades principales que realizará la aplicación | **0** |
|  | **** |
|  |  | **Necesita Mejorar:** Está basada en los requisitos no funcionales del sistema, incluyendo performance, disponibilidad, concurrencia, integridad, tolerancia a fallos. Define procesos, en forma básica, en las actividades o funciones principales del sistema. | **2** |
|  | **[4.0 / 20.0]** | **** |
|  |  | **Esperado:** Está basada en los requisitos no funcionales del sistema: performance, disponibilidad, concurrencia, integridad, tolerancia a fallos. Define procesos identificando: los mecanismos de comunicación, los servicios síncronos o asíncronos que serán utilizados, la llamadas a procedimientos remotos, difusión de eventos, etc. para las actividades o funciones principales del sistema. | **4** |
|  |  | **** |
|  | **C1.3 Vista de Desarrollo** | **Insuficiente:** No muestra la totalidad de módulos que van a ser desarrollados por el sistema | **0** |
| **** |
|  |  | **Necesita Mejorar:** Se han definido subsistemas o paquetes. Los subsistemas se han organizado en capas, cada una de las cuales tiene bien definido las interfaces de comunicación hacia las capas superiores. | **2** |
| **[5.0 / 20.0]** | **** |
|  |  | **Esperado:** Se han definido subsistemas o paquetes. Los subsistemas se han organizado en capas, cada una de las cuales tiene bien definido las interfaces de comunicación hacia las capas superiores. Se ha tomado en cuenta el reuso de componentes, las limitaciones del lenguaje utilizado, la portabilidad de la aplicación y la seguridad de la misma. | **5** |
|  | **** |
|  | **C1.4 Vista Física**  **[4.0 / 20.0]** | **Insuficiente:** No toma en cuenta los requisitos no funcionales del sistema como la disponibilidad, confiabilidad y escalabilidad. | **0** |
|  | **** |
|  | **Necesita mejorar:** Identifica de manera incompleta los componentes de la red o nodos sobre los cuales van a ejecutarse los componentes de la aplicación, | **2** |
|  | **** |
|  | **Esperado:** Identifica y mapea todos los componentes de la aplicación sobre una red de computadoras o nodos de procesamiento. El mapeo debe ser flexible y no estar ligado al código fuente de la aplicación, de forma que se pueda cambiar de un entorno de pruebas a otro de producción rápidamente. | **4** |
|  | **** |
|  | **C1.5 Definición de Escenarios** | **Insuficiente:** No define escenarios para ver la interacción de las vistas. | **0** |
|  | **** |
|  |  | **Necesita Mejorar:** Define escenarios básicos para ver el trabajo conjunto de las cuatro vistas detalladas anteriormente. | **2** |
|  | **[4.0 / 20.0]** | **** |
|  |  | **Esperado:** Define escenarios relevantes (requerimientos principales de la aplicación) para ver el trabajo conjunto de las cuatro vistas detalladas anteriormente. Los escenarios identificados incluyen secuencias de interacción entre objetos y procesos | **4** |
|  |  | **** |

1. La rúbrica Cero tiene que ver con el nivel de calidad básica del documento: ortografía, sintaxis, coherencia, presentación y otros. Esta rúbrica es binaria: si el documento cumple, pasa a revisarse; caso contrario, el documento es rechazado. [↑](#footnote-ref-1)