



ITLA

Instituto Tecnológico de Las Américas

Consideraciones y Criterios de la Actividad a Realizar

Definición, Origen, Características, Tipos

Francisco Javier Batista Acosta

2021-0952

### ➤ **Orígenes de Datos o Data Source a Nivel de Software**

Un origen de datos es una combinación de un origen de datos y la información de conexión necesaria para obtener acceso a dichos datos. Algunos ejemplos de orígenes de datos son SQL Server, Oracle RDBMS, hojas de cálculo y archivos de texto. (support.microsoft.com) Hay dos tipos de orígenes de datos: orígenes de datos de máquina y orígenes de datos de archivos. Aunque ambos contienen información similar sobre el origen de los datos, difieren en la forma en que se almacena esta información. Debido a estas diferencias, se usan de maneras algo diferentes. DataSource es una función de texto que devuelve el nombre de aplicación, de base de datos o de tabla de alias de una cuadrícula. Utilice DataSource en un cuadro o en una celda de texto, o en el encabezado de una fila o columna. GridName; (Necesario) Cualquier cuadrícula del informe, entre comillas. Axis; (Opcional) Puntero de un segmento de diseño de la cuadrícula. El eje puede ser una de las palabras clave siguientes: row, col o column (no distinguen entre mayúsculas y minúsculas). ID; (Necesario si se utiliza Axis) Indica el tiempo de diseño, número de fila o letra de columna desde donde se recupera la información de la conexión a base de datos. Si ID hace referencia a un ID de fila o columna no válido o inexistente, se usarán los valores predeterminados de la cuadrícula. Asimismo, se usarán estos valores si la cuadrícula no dispone de una conexión a base de datos secundaria. InfoType; (Necesario) Una de estas palabras clave: App devuelve el nombre de la aplicación. DB devuelve el nombre de la base de datos. Alias devuelve el nombre de la tabla de alias. Para que se devuelva el alias, la cuadrícula debe tener la tabla correspondiente activada.

### ➤ **Manipulación de Datos**

La manipulación de datos es el proceso de cambiar o alterar datos para hacerlos más legibles y organizados. Por ejemplo, puede ordenar los datos alfabéticamente para acelerar el proceso de búsqueda de información útil. 3 Por su función: en esta primera clasificación, se entiende que los datos que se manejen pueden pertenecer a uno de los tres tipos de funciones que tiene: lógicos, simbólicos o su simbólicos. Por su variabilidad: en este aspecto, los metadatos se pueden clasificar según la variabilidad de los mismos, en dos grupos. Por un lado, estarían los inmutables, que son los datos que no cambian independientemente de la parte del recurso que sea visible Por su contenido: esta última clasificación es la más usual. En este caso, se fraccionan los metadatos por su contenido.

## ➤ **Estándares de Desarrollo de Software**

Desarrollar un software significa construirlo simplemente mediante su descripción. Esta es una muy buena razón para considerar la actividad de desarrollo de software como una ingeniería. En un nivel más general, la relación existente entre un software y su entorno es clara ya que el software es introducido en el mundo de modo de provocar ciertos efectos en el mismo. (prezi.com) El marco de la metodología de desarrollo del software (también conocida como SDM) no emergió hasta los 60. Según Elliott (2004) el ciclo de vida del desarrollo de sistemas (SDLC) puede ser considerado el marco de metodología formalizado más antiguo por haber construido los sistemas de información. La idea principal del SDLC ha sido "perseguir el desarrollo de sistemas de la información de manera deliberada, estructurada y metódica, requiriendo cada etapa del ciclo de vida desde el comienzo de la idea hasta la entrega final del sistema para que sea realizado rígida y secuencialmente" dentro del contexto en el que el marco se aplique. El objetivo principal de este marco de metodología en los 60 era "para desarrollar sistemas empresariales funcionales a gran escala en una era de conglomerados de negocios a gran escala. Las actividades de los sistemas de la información evolucionaron alrededor del procesamiento de datos pesados y las rutinas de procesamiento de números".

## ➤ **Documentaciones y Comentarizaciones a Nivel de Software**

- La documentación de software puede estructurarse en uno de dos formatos, el manual de consulta y la guía de usuario. A veces, el mejor enfoque es utilizar una combinación de ambos formatos. Un manual de consulta se dedica a explicar las características individuales de un software (botón, barra, campo, y cuadro de diálogo) y cómo funcionan. (wikihow.com) 1. Documentación del proceso La documentación del proceso incluye todos los registros del desarrollo y mantenimiento del software. Los desarrolladores crean la documentación del proceso cuando desarrollan el concepto inicial y continúan agregando información relevante durante cada paso del proceso. Este tipo de documentación mantiene informado al equipo del desarrollador, guía a cada miembro del equipo a través de su papel en el proceso y ayuda a todos los involucrados a lograr los objetivos del proyecto de manera eficiente. 2. Documentación de planificación 3 Un plan de desarrollo de software (SDP) describe la razón del desarrollador para continuar con el proyecto y cómo pretende crearlo. También incluye métodos de desarrollo y brinda a los miembros del equipo un esquema para monitorear el proyecto mientras trabajan en él. Algunos factores que aborda un SDP incluyen: Problemas que resolverá el software El enfoque de desarrollo que utilizará el equipo Las funciones principales del software. El orden del desarrollo Roles de liderazgo para el proyecto. Las responsabilidades de cada miembro del equipo. 3. Estimar la documentación Antes de iniciar el proyecto, los desarrolladores de software elaboran estimaciones para incluirlas en la documentación del proceso. Una estimación, también conocida como estimación de esfuerzo, registra el tiempo o el dinero que el desarrollador espera que requiera el proceso. Las estimaciones de esfuerzo ayudan a los desarrolladores a planificar los presupuestos de los proyectos, elegir equipos y decidir los precios. 4. Documentación de normas Los estándares de software son normas establecidas, sistemas de reglas u otros formatos utilizados para crear software. Los estándares abiertos se refieren a aquellos a los que el público y otras empresas pueden acceder para crear

su propio software. Los estándares cerrados describen formatos y protocolos que el desarrollador no pone a disposición de otros, como procesos patentados. Los desarrolladores documentan los estándares de software de su proyecto para que su equipo comprenda los objetivos del proyecto y cómo abordarlos.

5. Documentación de métricas Las métricas de software registran los aspectos medibles del proceso de desarrollo y el producto. Las métricas de productos incluyen el tamaño, la complejidad y la calidad del software, mientras que las métricas de procesos miden la eficiencia de las técnicas y herramientas que utilizan los equipos para desarrollar software. Las métricas de software permiten a los desarrolladores: Comparar diferentes métodos de desarrollo. Preparar especificaciones de calidad. Medir la precisión de las estimaciones de esfuerzo Dar retroalimentación a los gerentes a lo largo del proceso de desarrollo

6. Documentación de programación Una vez que un desarrollador ha organizado un plan para su proyecto, usa un software especializado para hacer un cronograma detallado para sí mismo y para otros miembros de su equipo. El cronograma incluye las tareas que los miembros del equipo deben completar, los 3 plazos para terminirlas, el tiempo que el desarrollador estima que tomará cada tarea y una lista de recursos para cada miembro del equipo, si están disponibles. Los horarios juegan un papel importante para mantener a todos en el equipo al día durante todo el proceso de desarrollo.

7. Documentación del producto La documentación del producto describe el software que el equipo de desarrollo planea crear e instruye sobre cómo cambiarlo y usarlo. La documentación del producto consta de dos tipos, la documentación del sistema y la documentación del usuario. El equipo de desarrollo utiliza la documentación del sistema, mientras que los usuarios finales se benefician de la documentación del usuario. Ambos son importantes porque detallan los componentes del producto y cómo los usuarios pueden aplicarlos para resolver un problema.

8. Documentación del sistema La documentación del sistema describe los componentes del software y cómo funcionan juntos para producirlo. Principalmente beneficia a los miembros del equipo de desarrollo al permitirles realizar un seguimiento del progreso, modificar elementos específicos y predecir cómo funcionará el software. La documentación del sistema incluye muchos aspectos del software, incluidos: Capacidades Diseño Limitaciones Operación Mantenimiento

9. Documentación de requisitos La documentación de requisitos de software describe lo que el producto necesita para funcionar de manera eficaz. Los desarrolladores suelen redactar un documento de requisitos durante las etapas de planificación del desarrollo porque puede ayudar a guiar al equipo a través de la creación del software. El documento de los desarrolladores de requisitos de software incluye especificaciones de hardware, requisitos de funcionalidad y compatibilidad.

10. Arquitectura y documentación de diseño. La documentación de arquitectura y diseño proporciona información básica sobre cómo funciona el software. Brinda a los profesionales de la experiencia del usuario una descripción general del propósito del software, sus componentes y cómo funciona. Esta información les permite garantizar que el software tenga un diseño lógico con un flujo de datos controlado que brinde al usuario final una experiencia positiva.

11. Documentación técnica La documentación técnica describe cómo el software utiliza o se basa en conceptos tecnológicos existentes. Como sugiere su nombre, está dirigido a especialistas en tecnología de la información (TI) e ingenieros de software que lo utilizan para garantizar una 3 experiencia de usuario de calidad y ganarse la confianza y la lealtad de los clientes. La documentación técnica incluye elementos como rutas de interfaz de protocolo de aplicación (API), que permiten que el software se comunique con el dispositivo del usuario, y kits de

desarrollo de software (SDK), o un conjunto de herramientas que utilizan los desarrolladores para crear software.

12. Documentación del usuario La documentación del usuario incluye los recursos proporcionados a los usuarios finales y administradores del sistema para enseñarles cómo utilizar el producto. También puede incluir características especiales de su software, consejos de uso y consejos para la resolución de problemas. Los tipos comunes de documentación del usuario incluyen preguntas frecuentes, tutoriales e información de soporte.

13. Documentación del usuario final La documentación del usuario final se refiere a la información proporcionada a las personas que utilizan un producto de software finalizado. Estos recursos facilitan a los usuarios navegar y familiarizarse con el software. Los tipos de documentación para el usuario final incluyen: Requisitos del sistema operativo Instrucciones de instalación Cómo iniciar y utilizar el software Fotos de las características del software Consejos para solucionar problemas Información de contacto del servicio de atención al cliente

14. Documentación del administrador del sistema La documentación del administrador del sistema ayuda a los responsables de administrar los sistemas informáticos y los servidores a mantener el software. Por lo general, incluye información sobre la instalación, las actualizaciones y la funcionalidad del software. También puede proporcionar una guía sobre cómo se comporta el software con diferentes sistemas y qué hacer si no funciona correctamente.

### **Resumen Crítico**

- **Bueno en un resumen de todo vemos como se aplica todo lo aprendido e investigamos para aplicarlos en el proyecto final, aclarar dudas y nos sirve para proyectos y trabajos futuros.**

### **Bibliografía**

buom.store

wikipedia.or

powerdata.es