

**NOMBRE DEL PROFESOR: Juan José Ojeda Trujillo**

**NOMBRE DEL ALMUNO: marlong Uriel ramos dominguez**

**MATERIA: Ingeniería en software**

**CARRERA: Ingeniería en sistemas computacionales**

**CUATRIMESTRE: 8**

**Un componente es una parte física de un sistema (modulo, base de datos, programa ejecutable, etc.). Se puede decir que un componente es la materialización de una o más clases, porque una abstracción con atributos y métodos pueden ser implementados en los componentes.**

**El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.**

**4.1 Modelos de implementación**

4.2DIAGRAMAS DE COMPONENTES

**• Componentes • Agregación**

**• Interfaces • Realización**

**• Asociación • Asociación**

**4.3 ELEMENTOS DEL DIAGRAMA DE COMPONENTES**

**El Diagrama de Despliegue es un diagrama que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.**

**4.4 DIAGRAMAS DE DESPLIEGUE**

• **Pruebas en que se conoce el código a probar**

**• Algunas clases de pruebas**

**• Pruebas de condiciones**

v

v

**• Encontrar defectos en el software**

**• Pruebas de Verificación**

**• Pruebas de Validación**

**Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos el cual gestiona los datos que debe utilizar la aplicación; en líneas generales del MVC dicha gestión corresponde al modelo**

**Para crear un objeto se necesita reservar suficiente espacio en memoria e inicializar los valores de los campos que representan el estado del objeto. Este trabajo es realizado por un tipo especial de método denominado constructor.**

**Los casos de uso son una técnica para la especificación de requisitos funcionales propuesta inicialmente por Ivar Jacobson [Jacobson, 1987], [Jacobson et al. 1992] e incorporada a UML Modela la funcionalidad del sistema tal como la perciben los agentes externos, denominados actores, que interactúan con el sistema desde un punto de vista particular**

**Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones**

**Cuando se va a construir un sistema software es necesario conocer un lenguaje de programación, pero con eso no basta. Si se quiere que el sistema sea robusto y sustentable, es necesario que el problema sea analizado y la solución sea cuidadosamente diseñada. Se debe seguir un proceso robusto, que incluya las actividades principales**.

4.10 MVC Y BASES DE DATOS

4.9 ARQUITECTURA LÓGICA CON PATRONES: SEPARACIÓN MODELO- VISTA

**4.8 REALIZACIÓN DE LOS CASOS DE USO Y DIAGRAMAS DE INTERACCIÓN**

**4.7 Constructores y destructores declaración, uso y aplicaciones**

4.6 IMPLEMENTACIÓN EN JAVA DE LOS DIAGRAMAS DE CLASE

**4.5 MODELOS DE PRUEBA**

UNIDAD IV MODELO DE IMPLEMENTACIÓN

REPORTE DEL BRAZO ROBOTICO



