

**Instituto Tecnológico Superior Zacatecas Occidente**

**Ing. Sistemas Computacionales**

**Practica**

## “TACTICAS DE ATRIBUTOS DE CALIDAD”

****

**Arquitectura y Diseño de Software**

* **Alma Yadira Esparza Montelongo**
* **Gabriela Quintanar Chávez**
* **Brenda Gómez Salas**

**Erika Jazmín Robles Gómez**

**10 de marzo de 2015**

**TACTICAS DE ATRIBUTOS DE CALIDAD**

**Seguridad**

Tiene como propósito mantener a los componentes del sistema y los datos de este en un estado coherente.

Las tácticas de seguridad pueden ser divididas de acuerdo a los objetivos principales restringir ataques detectar ataques y recuperarse de ataques.

**Resistencia a los ataques:** Su principal objetivo de estas tácticas es no permitir el acceso de usuarios no deseados al sistema proporcionando mecanismos de protección de confidencialidad de la información las principales aplicaciones de software son las Siguientes:

**Autentificación de usuario:** Garantiza que el usuario o computadora remota tengan acceso a funciones del sistema con contraseñas o certificados digitales

**Autorización de usuario:** Garantiza que el usuario autentificado tiene los derechos para acceder y modificar los servicios del sistema.

**Mantenimiento de confidencialidad de datos.** Garantiza la protección de datos de usuarios no autorizados. Es usualmente lograda implementando alfana forma de encriptación a los datos o enlaces de comunicación.

**Usabilidad**

Las tácticas de usabilidad se enfocan en facilitarle al usuario actividades que tenga que realizar con el sistema. Dos de los tipos de tácticas de usabilidad empleadas son:

**Tácticas de tiempo de ejecución.** Este tipo de tácticas consiste en ayudar al usuario a realizar sus actividades fácilmente. En genera consiste en retroalimentar al usuario sobre lo que hace el sistema y permite realizar operaciones tales como cancelar, deshacer, etc.

**Tácticas de tiempo de diseño:** Principalmente este tipo de tácticas consisten en decisiones de diseño que permitan al usuario interactuar con el sistema. Se implementan con patrones de MVC (Modelo Vista Controlador), mismo que es utilizado en la táctica de coherencia semántica del atributo de calidad de modificabilidad.

**Mantenibilidad**

Utilizaremos un pratron de modelo vista controlador (MVC)

**FRAMEWORK CAKEPHP**

CakePHP es un framework para el desarrollo aplicaciones web escrito en PHP, creado sobre los conceptos de Ruby on Rails.

CakePHP o pastelPHP es un framework o marco de trabajo que facilita el desarrollo de aplicaciones web, utilizando el patrón de diseño **MVC**(**M**odelo **V**ista **C**ontrolador). Es de código abierto y se distribuye bajo licencia MIT.

Al igual que Ruby On Rails, CakePHP facilita al usuario la interacción con la base de datos mediante el uso de ActiveRecord. Además hace uso del patrón Modelo Vista Controlador.

* Compatible con PHP4 y PHP5
* CRUD de la base de datos integrado
* URLs amigables
* Sistema de plantillas rápido y flexible
* Ayudas para AJAX, Javascript, HTML, forms y más
* Trabaja en cualquier subdirectorio del sitio
* Validación integrada
* Scaffolding de las aplicaciones
* Access Control Lists
* Sintetización de datos
* Componentes de seguridad y sesión

**PATRONES DE DISEÑO EN CAKEPHP**

CakePHP implementa diferentes patrones de diseño, definidos por la wikipedia como:

Los patrones de diseño (design patterns) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño es una solución a un problema de diseño.

Entre otros, CakePHP implementa los siguientes patrones:

Singletón

El patrón de diseño :

El patrón de diseño singleton (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

En el artículo Patron Singleton en PHP5 puedes encontrar una explicación sencilla y clara.

En CakePHP la clase Configure actúa como un singletón, y sus métodos pueden ser llamados desde cualquier lugar de la aplicación, en un contexto estático.