

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
EXTENSIÓN LATACUNGA

MAESTRÍA EN INGENIERÍA DE SOFTWARE

TERCERA PROMOCIÓN

TÍTULO DEL PROYECTO:

"MARCO DE TRABAJO PARA DESARROLLO RÁPIDO DE
APLICACIONES WEB ASÍNCRONAS DE TIEMPO REAL PARA
EL ÁREA DE DESARROLLO DE SOFTWARE DEL
DEPARTAMENTO DE TICS DE LA ESCUELA POLITÉCNICA
DEL EJERCITO EXTENSIÓN LATACUNGA"

Autor:

Ing. Raúl Armando Villamarín Yagual.

INTRODUCCIÓN

Web de Tiempo Real

- Tecnologías
- Transmisión instantánea
- Comunicación sw-Usuario

Realidad Nacional

- No explotado
- Innovación
- Complejidad

PROBLEMA

Web de Tiempo Real

- Falta de Soporte
- Esfuerzo
- Tecnologías Ambiente web

TICs ESPE

- Modelo Actual CMS
- RTW no aprovechado
- Complejidad - Limitante



SOLUCIÓN

Marco de Trabajo

- Soporte RTW
- Desarrollo Rápido
- Patrones
- Estructura
- Componentes
- Flexible

TICs ESPE

- Explotación Tecnologías
- RTW
- Innovación
- Micrositios

PROYECTO E IMPORTANCIA

Proyecto

- Evolución Web
- Investigación patrones
- Buenas prácticas
- Integración RTW
- Caso Práctico

Importancia

- Progreso Tecnologías
- Inexistencia Marco Trabajo
- Tecnología no explotada

OBJETIVOS E HIPÓTESIS

Objetivos

- Construir
- Determinar estado arte
- Elaborar, Crear
- Aplicar Caso Práctico
- Validar

Hipótesis

- Si se construye MT RTW
- Se reduce complejidad



Variables

Variable
Dependiente
(Indicadores)

- Sencillez
- Claridad
- Delimitación
- Independencia
- Modularidad
- Expansibilidad
- Rapidez

Variable
Independiente

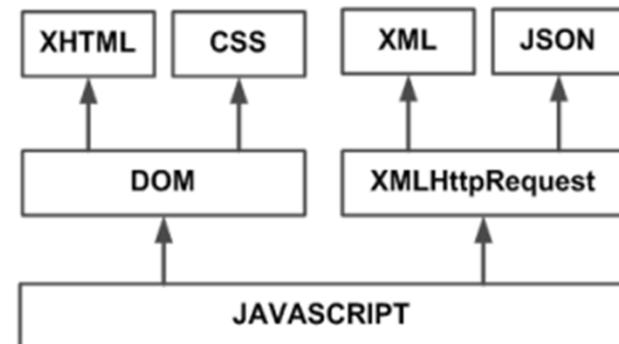
- Construcción del Marco de Trabajo

FUNDAMENTOS

Patrón mvc

- Modelo - Negocio
- Vista - Presentación
- Controlador – Gestión eventos

Web asincrónica



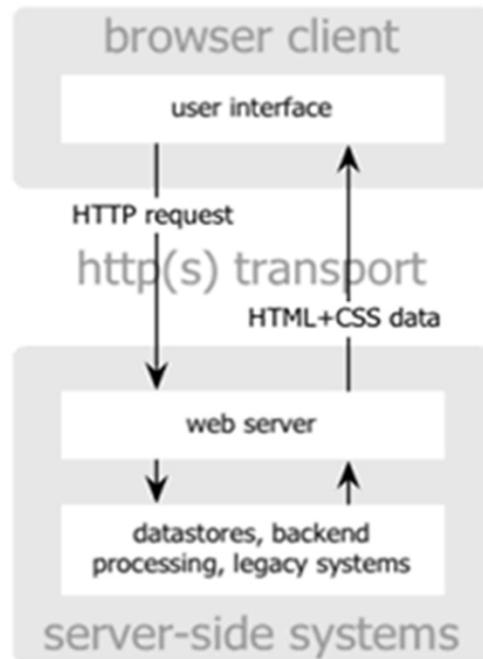
Push Comet

- Ajax inverso
- Modelo ligero
- Evolución Web

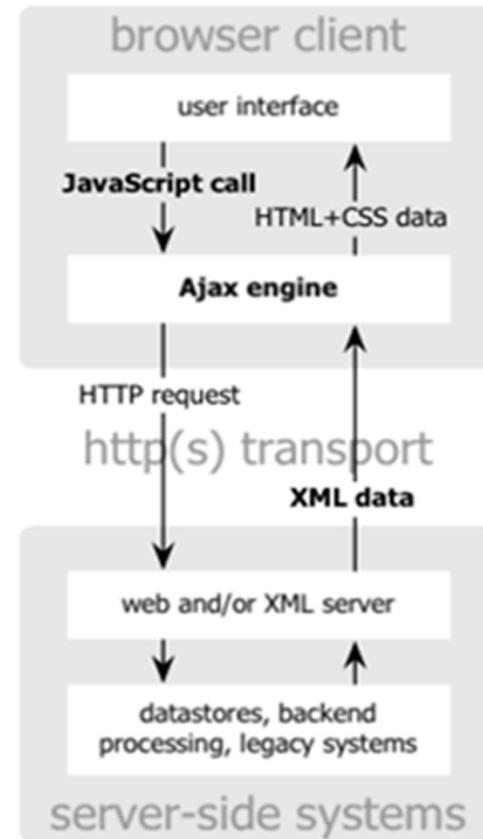
RTW

- Paradigma comunicación
- Instantáneo

FUNDAMENTOS



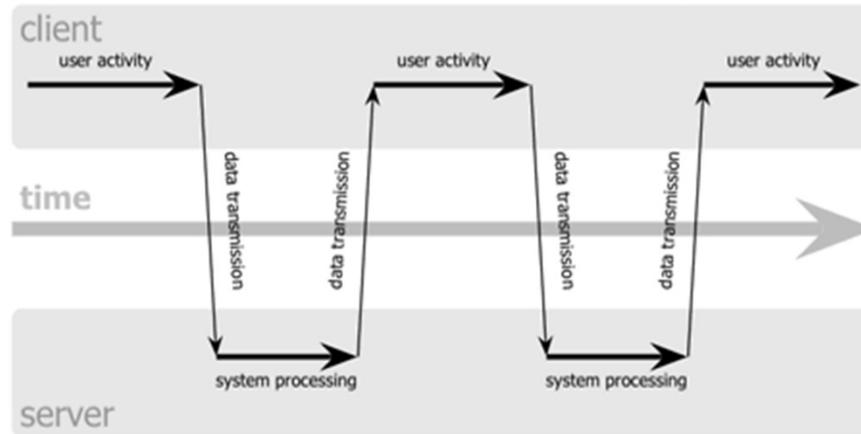
**MODELO DE APLICACIONES
WEB CLÁSICAS**



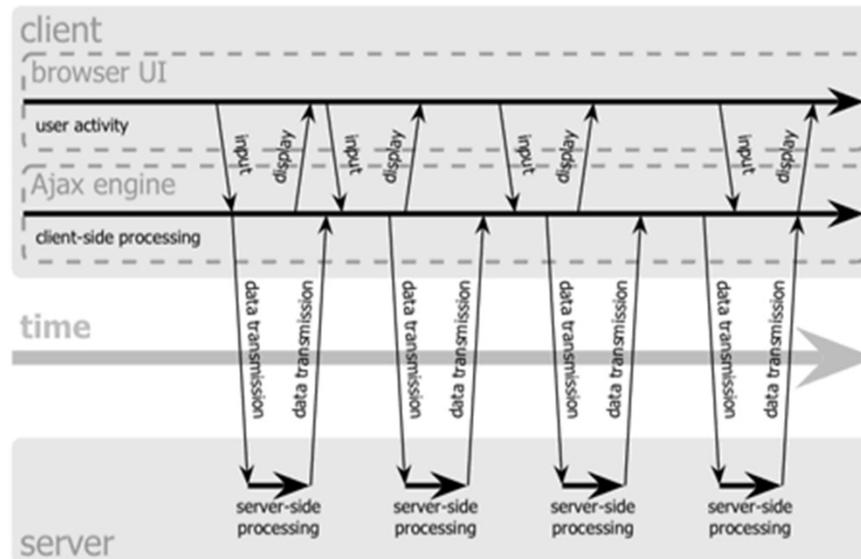
**MODELO DE APLICACIONES
WEB AJAX**

FUNDAMENTOS

aplicación web clásica (sincrónica)

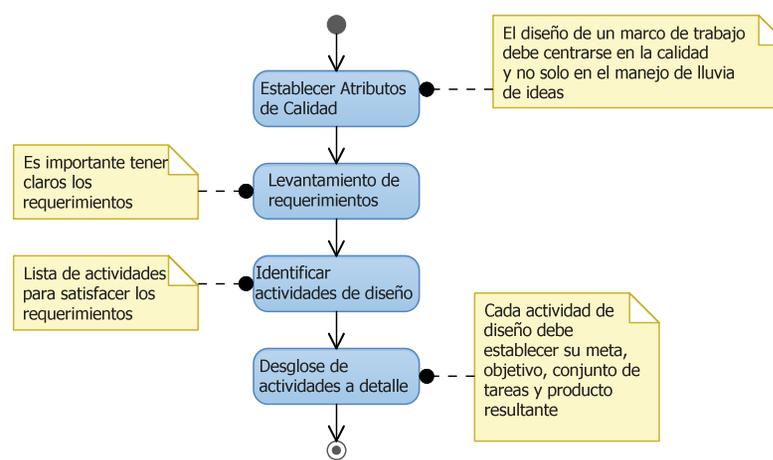


modelo de aplicación web ajax (asincrónica)



DISEÑO

Actividades



Atributos de Calidad

- Sencillez → Aprendizaje
- Claridad → Encapsulamiento
- Delimitación → Definido
- Expansibilidad → Funcionalidad
- Modularidad → Agrupación
- Independencia → Reuso
- Rapidez → Patrones

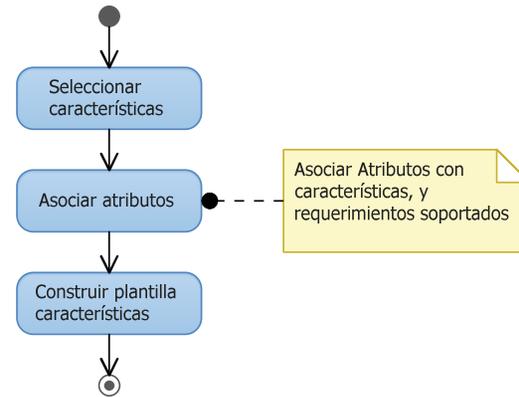
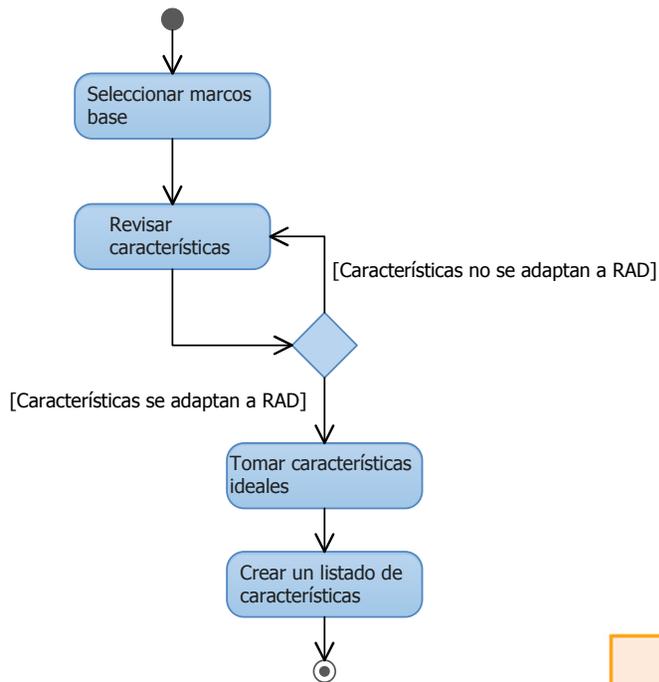


DISEÑO

Requerimientos

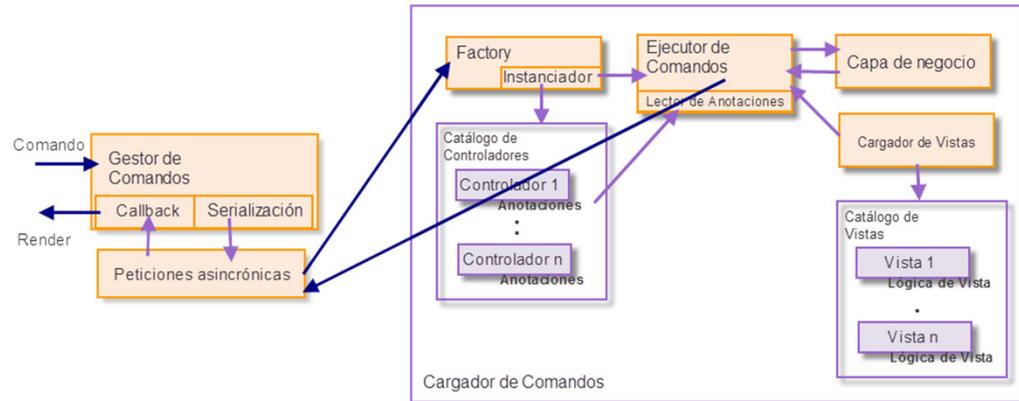
- Web
- Comet
- Mvc
- Encapsulamiento
- Orm
- Disparo Eventos
- Modelado Dominio
- Librerías
- Agrupación
- Reutilización
- Software libre
- Reusar librerías Open Source
- Núcleo Javascript
- Html
- Lenguaje Plantillas
- Patrón aplicación
- API

DISEÑO (Proceso)



DISEÑO

Modelo de peticiones Propuesto



Patrón de controlador propuesto

```
use_class('edu.bl.blEspecialidad');
use_class('administrativo.especialidades.ListaEspecialidadC');

/**
 * Description of ControladorFormulario
 *
 * @author raul
 * @view(especialidad)
 */
class ControladorFormulario {
    public $en_especialidad;
    private $blEspecialidad;

    public function __construct() {
        $this->en_especialidad = new especialidad();
        $this->blEspecialidad = new blEspecialidad($this->en_especialidad);
    }

    public function guardar() {
        $this->blEspecialidad->guardar();
        return new ListaEspecialidadC();
    }

    public function cargar($id) {
        $this->en_especialidad = $this->blEspecialidad->cargar($id);
        return $this;
    }
}
```

Importación de librerías u otros controladores

Anotación de nombre de vista con la que trabajará el controlador (asociación Controlador - Vista)

Clase de controlador. Las propiedades públicas tienen el mismo nombre del atributo id de la etiqueta en la vista

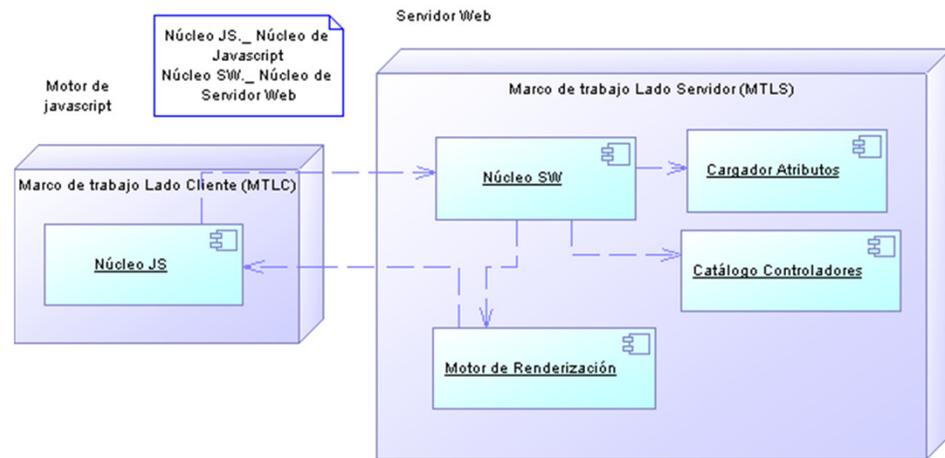
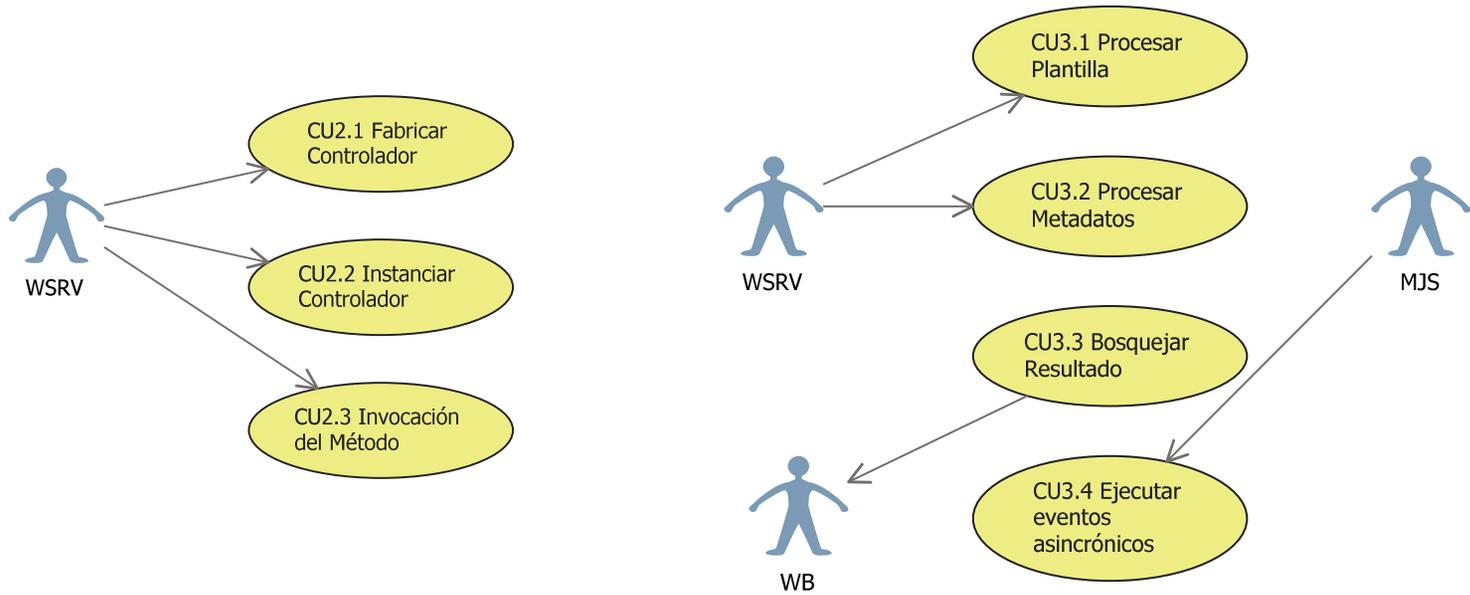
Variable de objeto de entidad. Implícitamente la clase de entidad es añadida importando la capa de lógica de negocio

Variable de objeto de lógica de negocio. La capa de lógica de negocio tendrá encapsulada las operaciones sobre entidad y contexto de la base de datos para la persistencia

Comandos. Los comandos son métodos de la clase de controlador para realizar operaciones sobre la capa de lógica de negocio

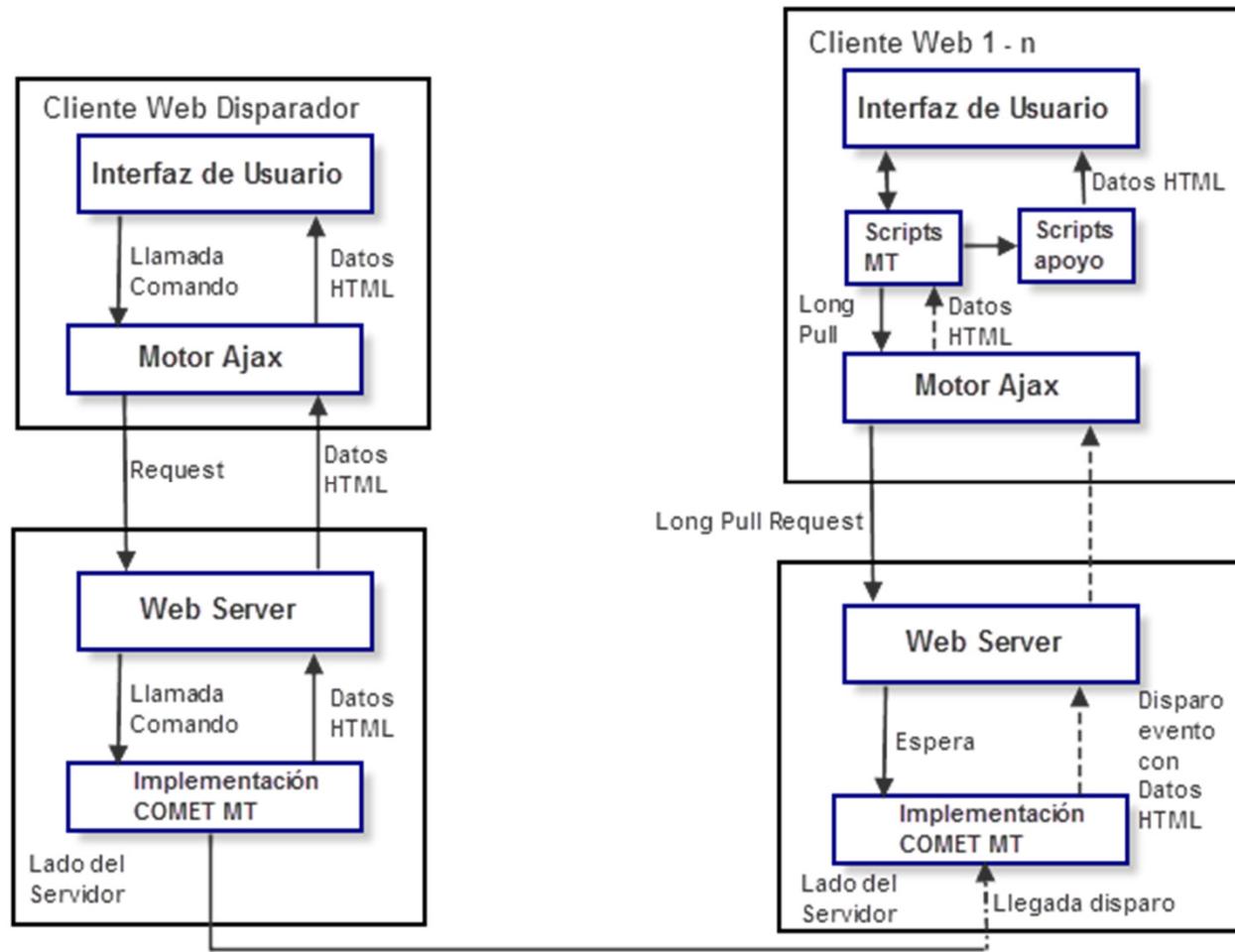
Retorno de Controladores. Los comandos (Métodos) del controlador pueden retornar otros controladores, facilitando la interacción entre controladores y asociaciones de vistas

DISEÑO



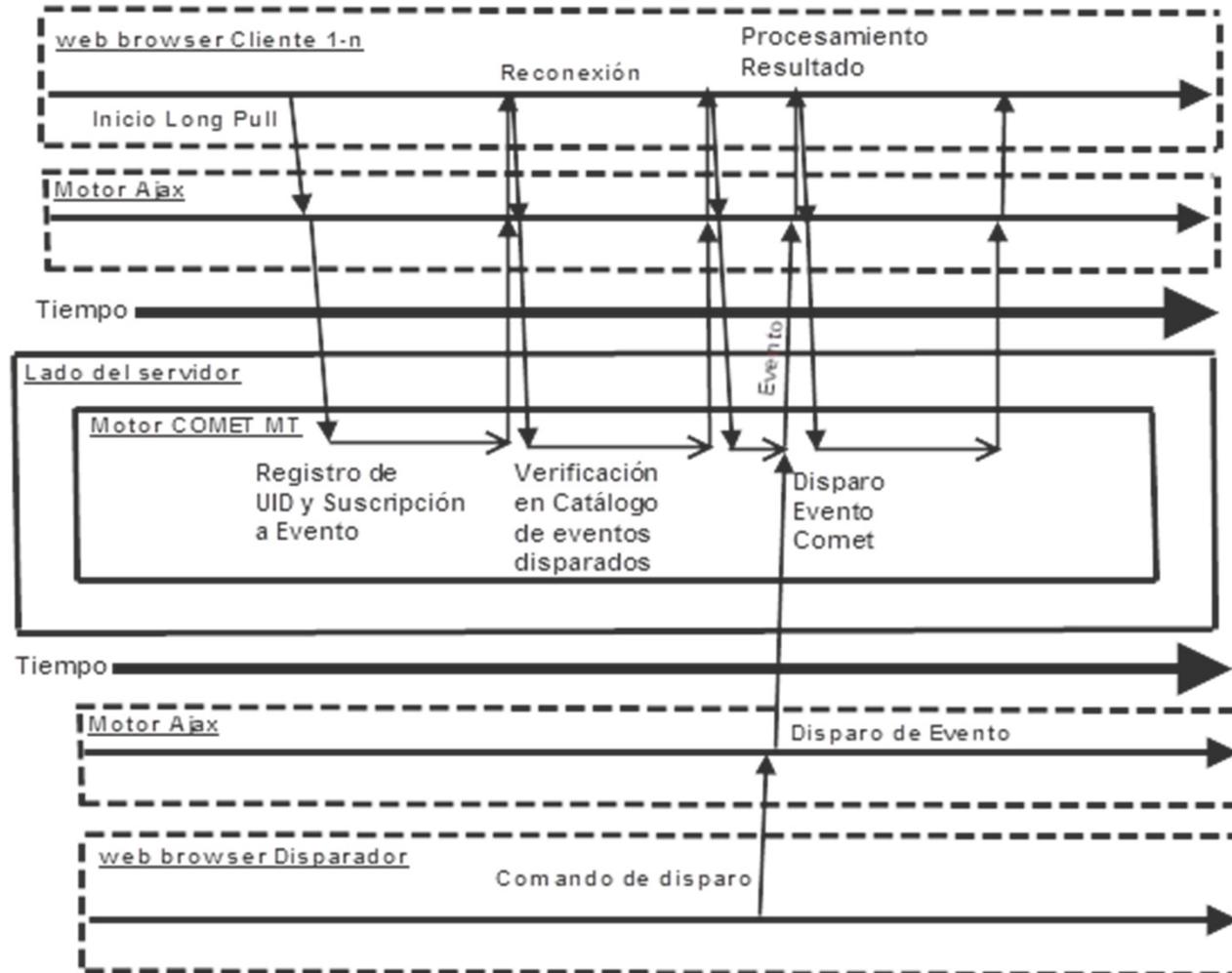
DISEÑO

Modelo de aplicación web de tiempo real



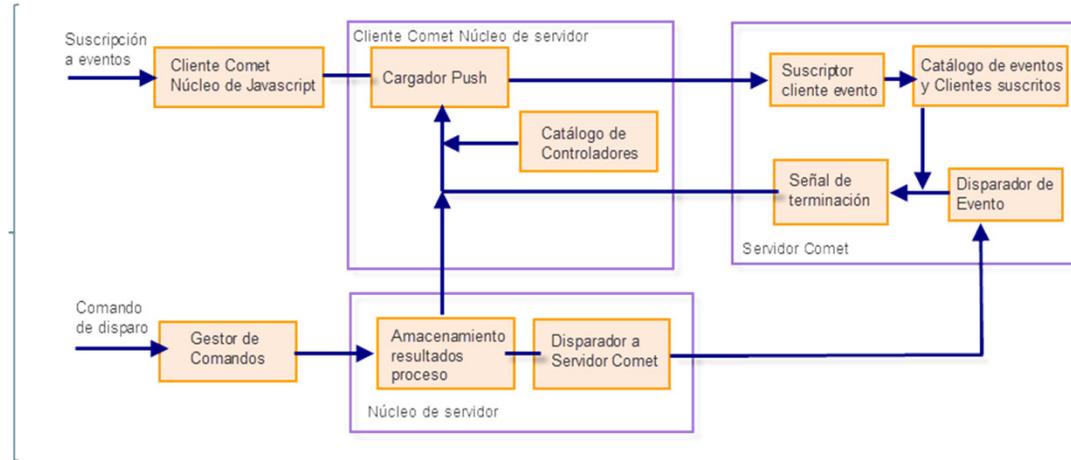
DISEÑO

Modelo de aplicación web de tiempo real



DISEÑO

Modelo de tratamiento Comet



Patrón de suscriptor

```

hnd_ Es una propiedad de identificador único.
pusher_ Es una propiedad que indica el controlador de eventos comet asincrónicos.
event_ Es la propiedad que indica el nombre del evento comet asincrónico, también podría establecerse un vector de eventos para manejar varios eventos con el mismo controlador.
render_ Es el id de un div en donde se dibujará el resultado del evento comet asincrónico habiéndose procesado por el motor de renderización de vista, es opcional

/*
 * Ejemplo de creación de suscriptor de eventos comet
 */
var MyHandler = Class.create(CometClientBase, {
  hnd: '1234',
  pusher: 'pruebas.pantdisparo.ControladorLlegadaComet',
  event: 'recibir',
  render: 'idDivARenderizarResultado',
  recibir: function(resultadoHtml, parametroAdicional1, parametroAdicional2) {
  }
});
var cometEjemplo = new MyHandler();
core.events.subscribePageReady(function() {
  cometEjemplo.start();
});

```

Definición de prototipo de suscriptor_ Personaliza el cliente suscriptor base

recibir_ Es el método del controlador de javascript en donde recibirá el resultadoHtml, y los parámetros adicionales que le haya enviado la vista renderizada, si render está utilizado se produce después de haberse dibujado el resultado

cometEjemplo_ Instancia del prototipo de manejador - Suscriptor MyHandler

Suscripción inicial en el evento de carga completa de la página, manejado por el núcleo de javascript

DISEÑO

Patrón de comando de disparo

```
/**  
 * Description of ControladorFormulario  
 *  
 * @author raul  
 * @view(ejemplo_disparador)  
 */  
class ControladorDisparador {  
  
    public function disparar() {  
  
        $datos_procesados = algunproceso();  
        //Suponiendo que datos procesados contenga un vector asociativo  
        //con la siguiente estructura:  
        //array('dato1' => 'resultado1', 'dato2' => 'resultado2')  
        //Básicamente se configura la llamada de la ejecución del evento  
        //el evento comet corresponderá al método recibir de la clase  
        //ControladorDisparo manteniendo una estructura como se indica:  
        //módulo.pantalla.Controlador.método  
        CometTrigger::triggerEvent(  
            'pruebas.pantdisparo.ControladorLlegadaComet.recibir', $datos_procesados  
        );  
    }  
}
```

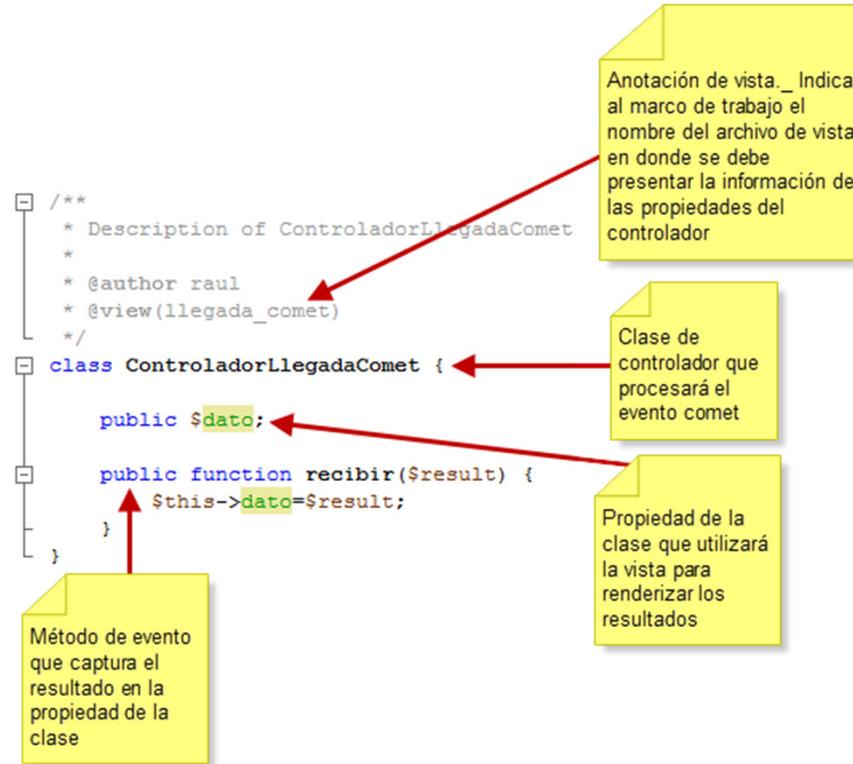
Ubicación de la clase de controlador de evento comet

Nombre de clase de controlador que procesará el evento comet

Nombre del método del controlador que receptorá los datos procesados

DISEÑO

Patrón de controlador de recepción

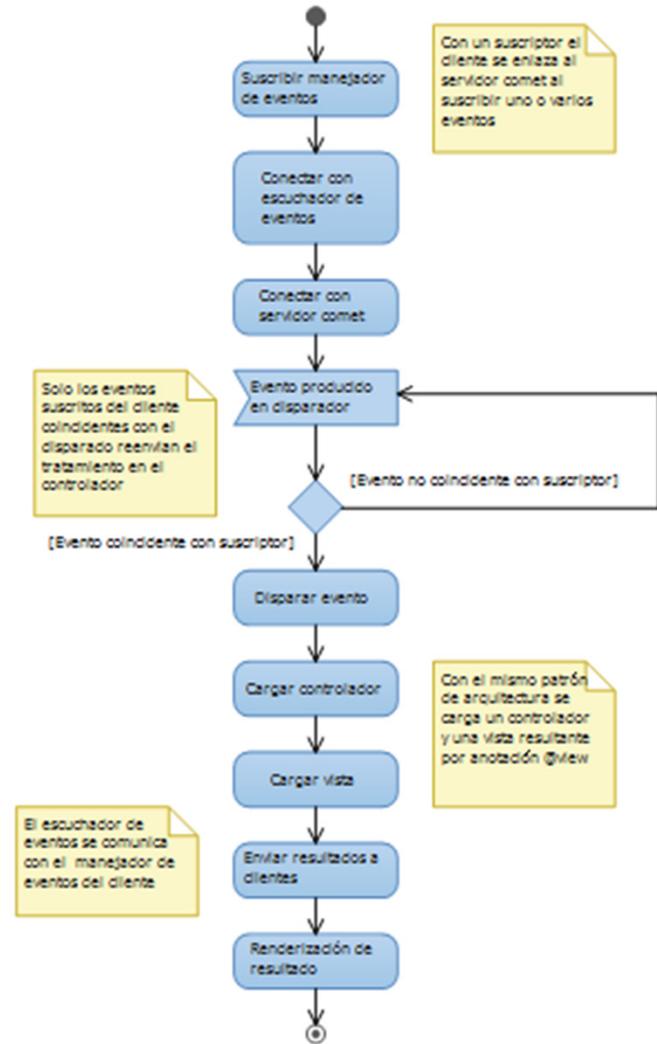
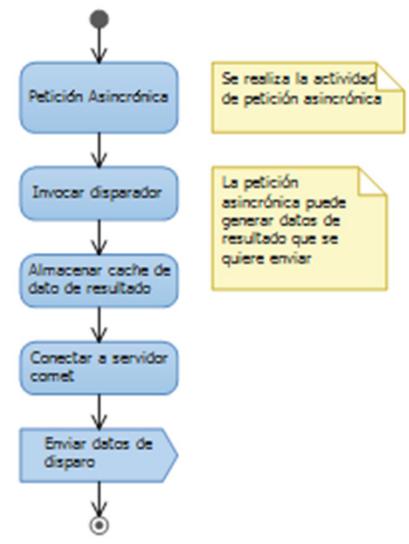
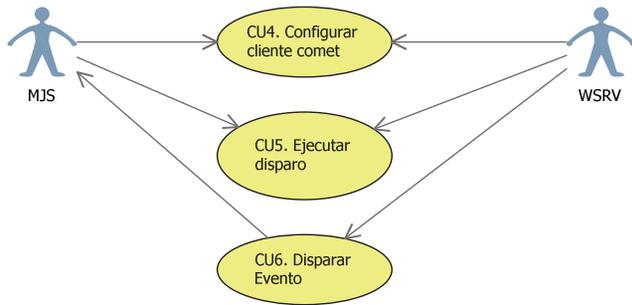


Patrón de vista

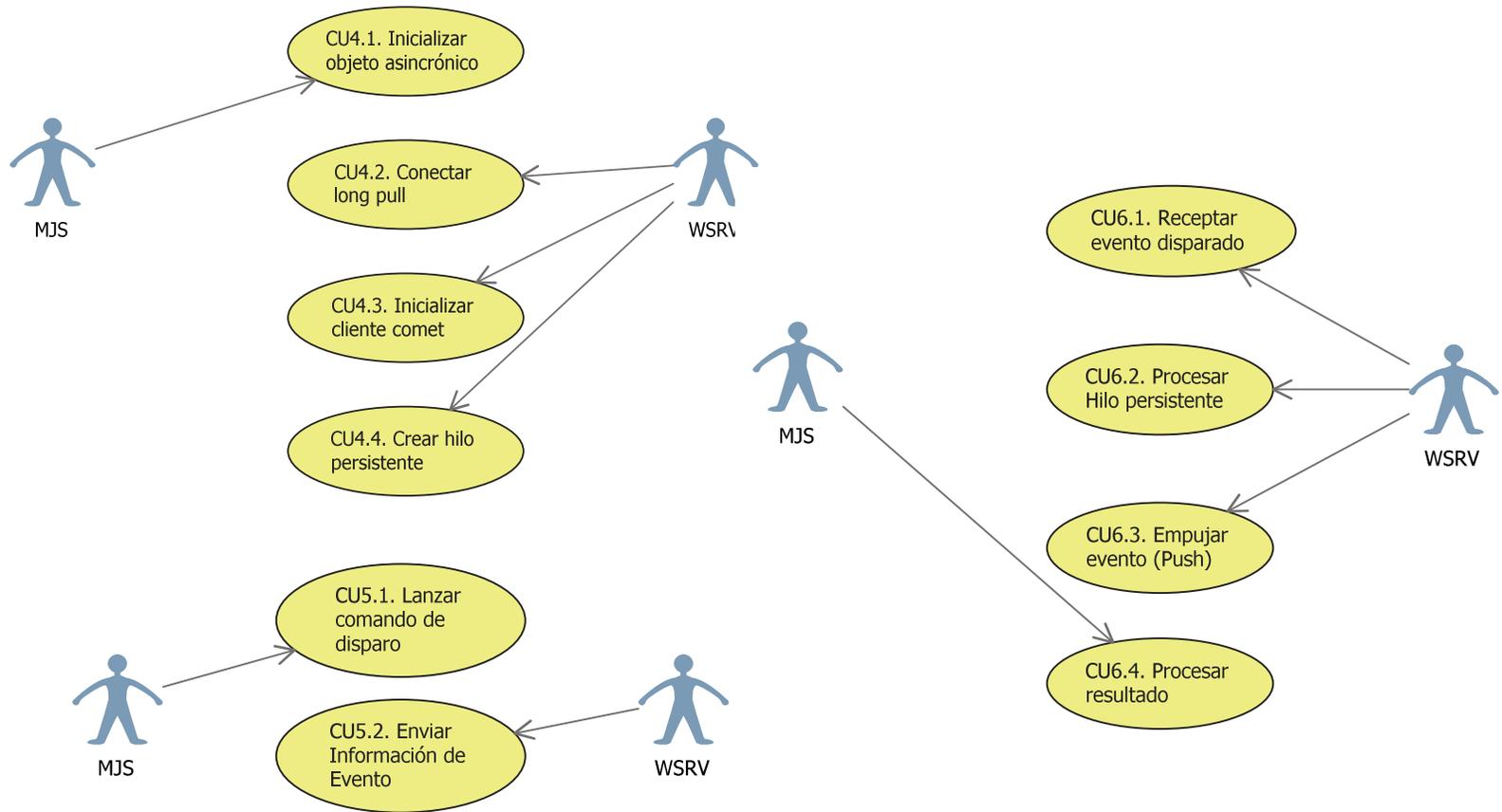


DISEÑO

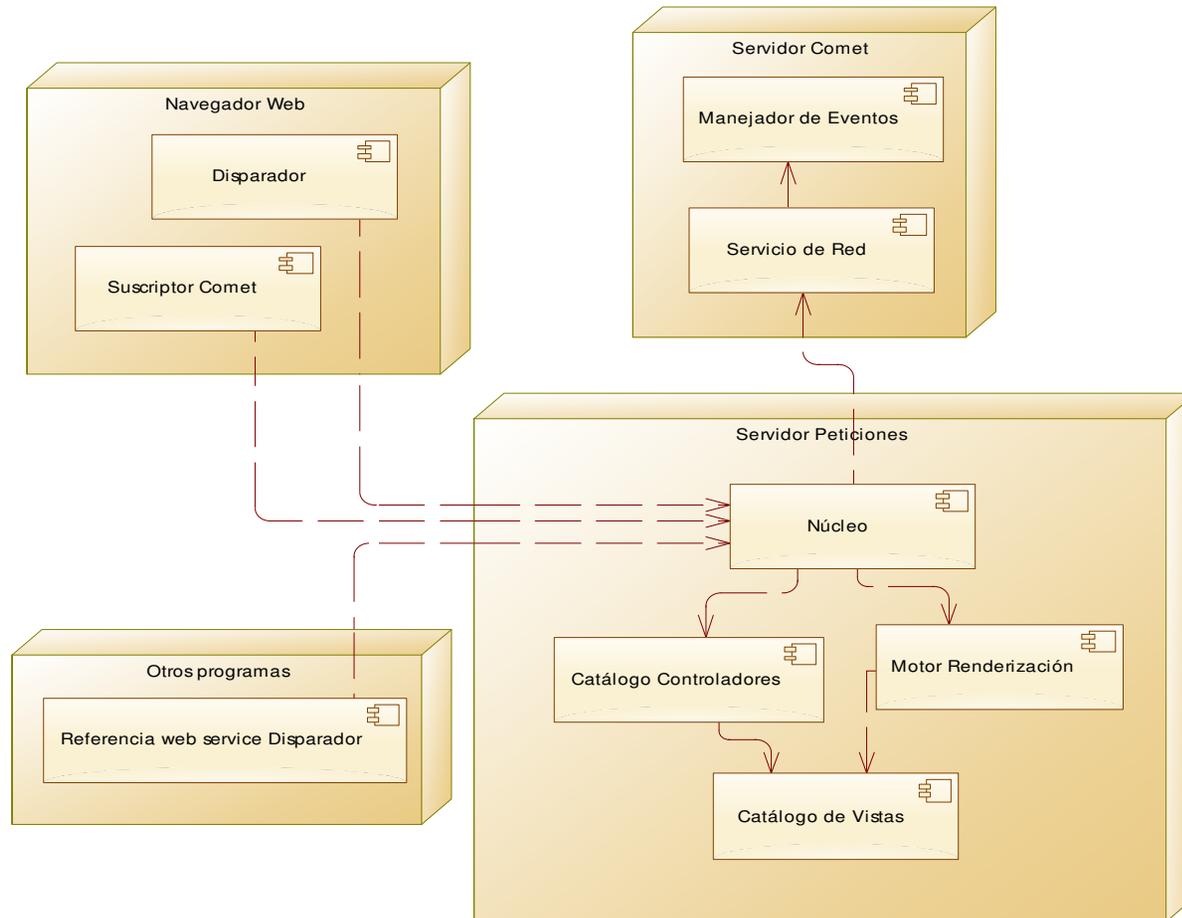
WB_ Web browser
 MJS_ Motor de Javascript
 WSRV_ Servidor Web



DISEÑO

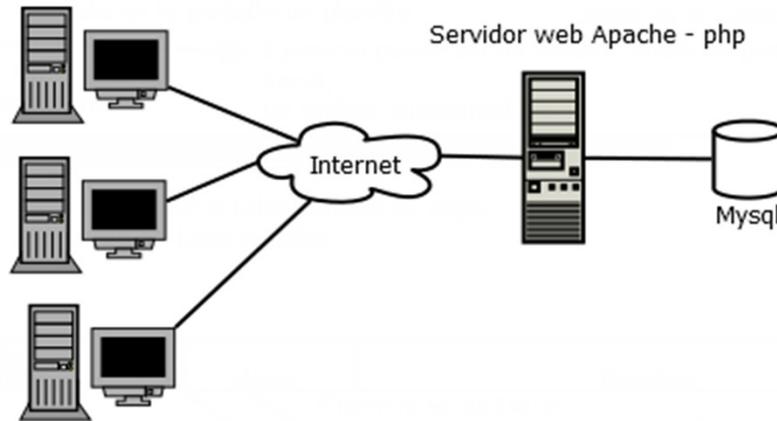


DISEÑO



CASO DE ESTUDIO PRÁCTICO

Cientes / Navegadores Web

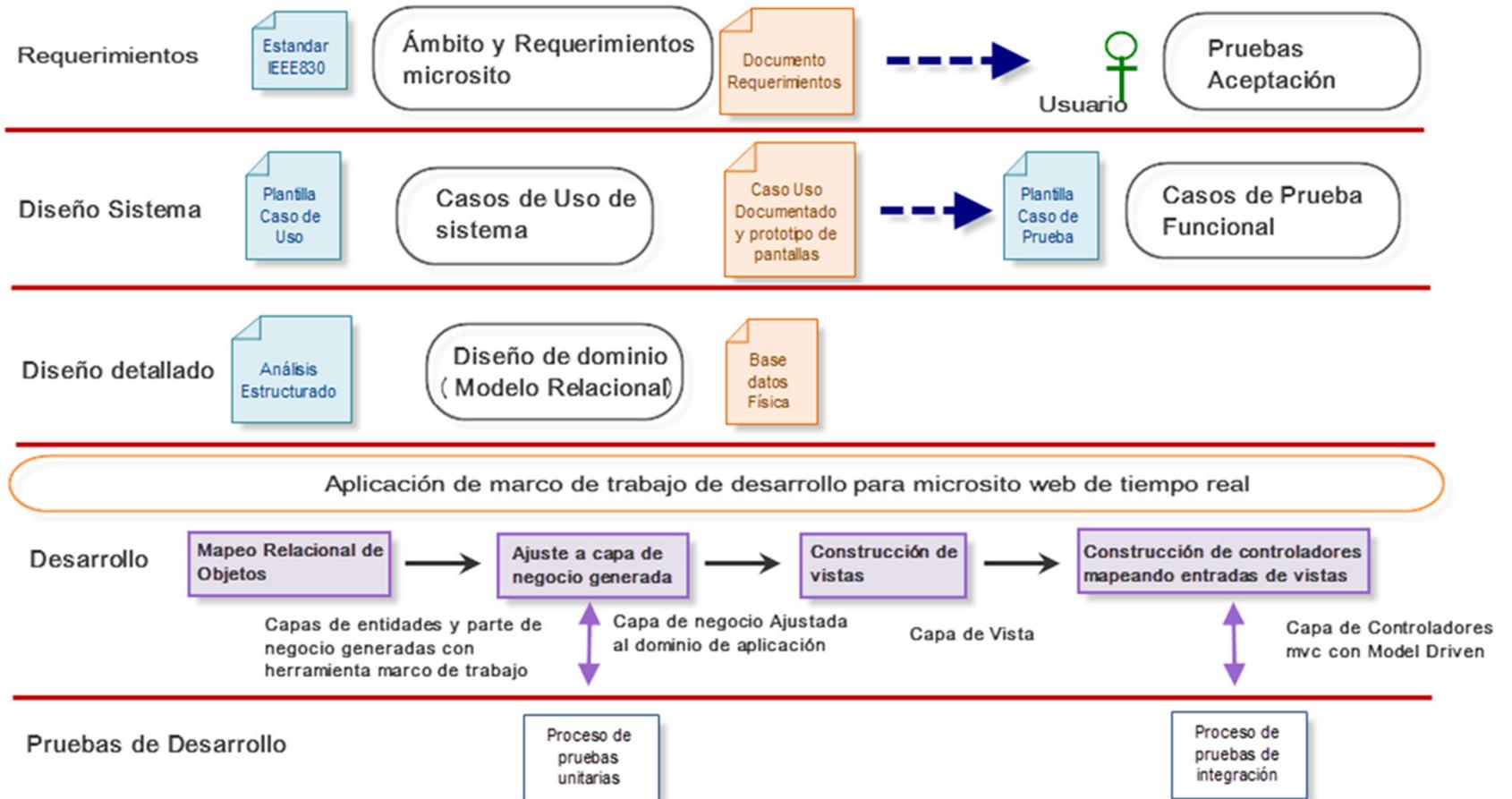


- Micrositio de Investigación
- Aplicación del marco de trabajo.
- Requerimientos de TICs
- Integración a CMS actual (Wordpress)

- Usuarios (Administrador, Investigador) .
- Proyectos.
- Publicaciones a tiempo real (Fotografías, Archivos).
- Muro de Publicaciones.

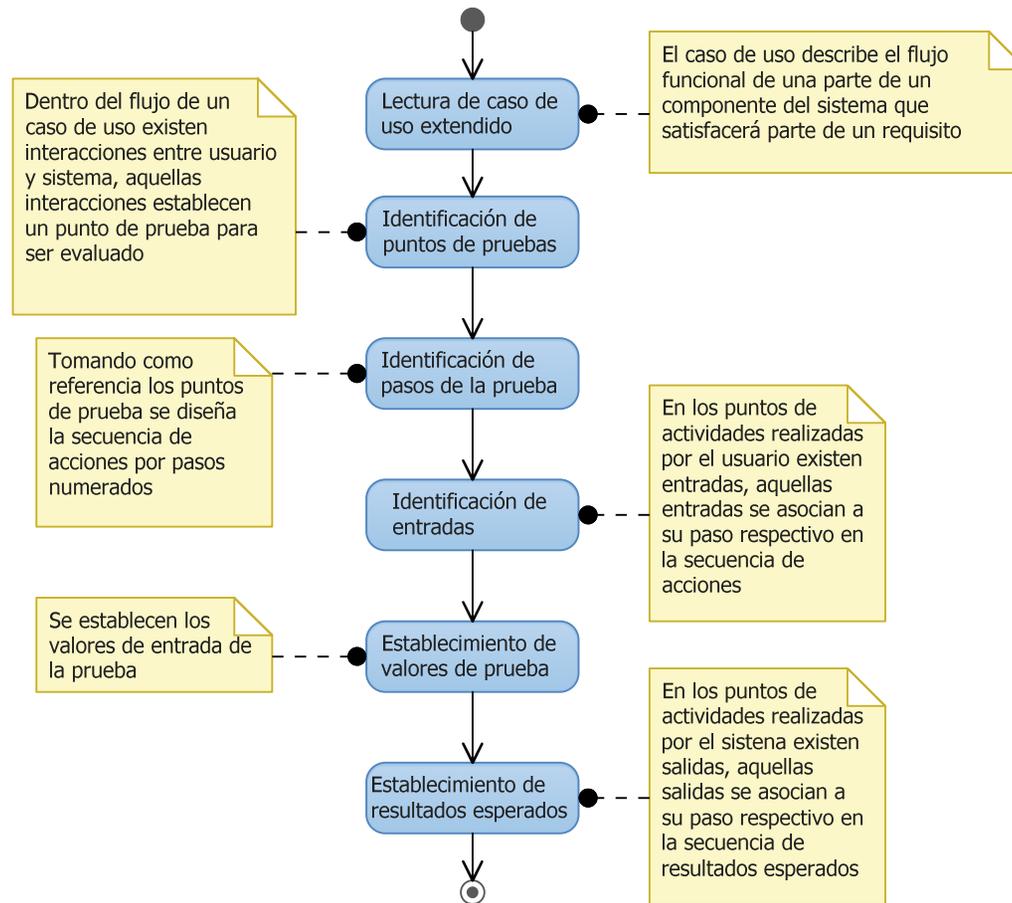
CASO DE ESTUDIO PRÁCTICO

Aplicación del Marco de trabajo - Ciclo de Vida en V



CASO DE ESTUDIO PRÁCTICO

Actividades para establecer casos de prueba

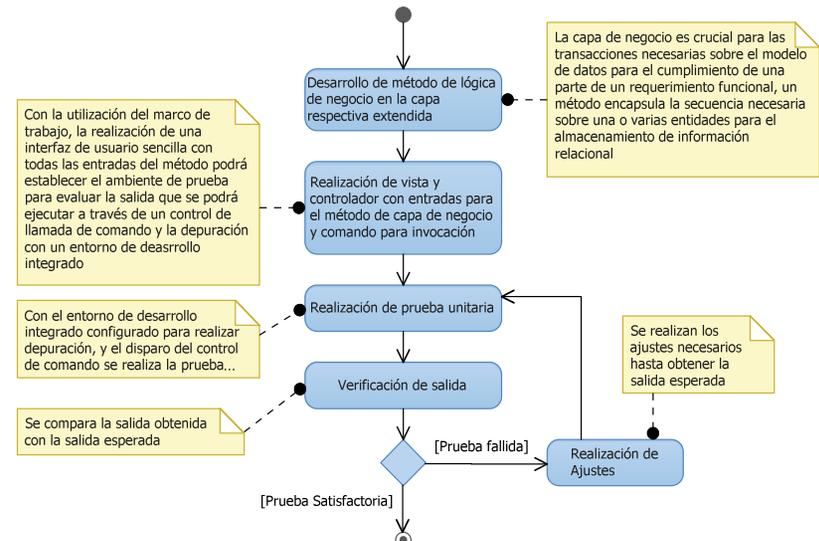


CASO DE ESTUDIO PRÁCTICO

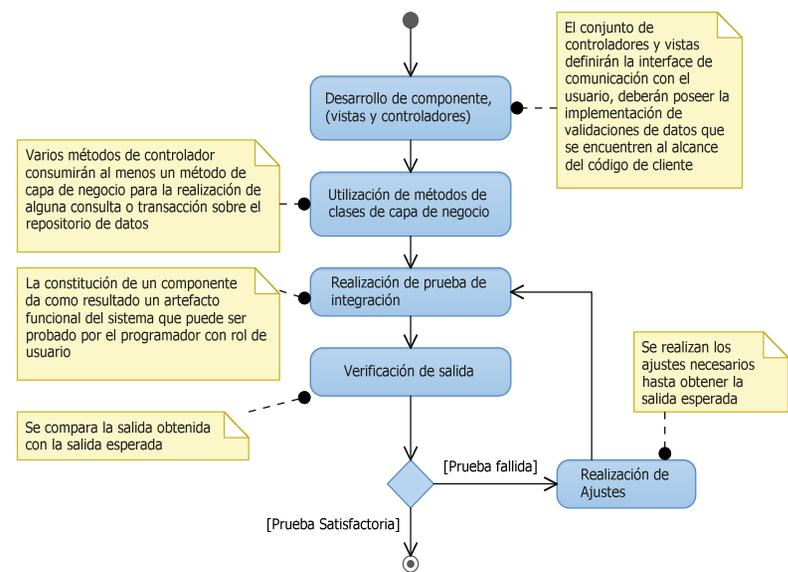
Nombre	[Nomenclatura y nombre del caso de prueba] Ejemplo CP01.Caso de prueba de Ingreso de datos de la pantalla de marcas.
Caso de Uso Relacionado	[Nomenclatura y nombre del caso de uso relacionado] Ejemplo CU01. Caso de uso de ingreso de marcas.
Objetivo	[Objetivo del caso de prueba a desarrollarse]
Acciones	[Secuencia de pasos y entradas] Ejemplo: 01->02->03(Entrada1,Entrada2)->04 [Pasos secuenciales] Ejemplo 1. La prueba selecciona el menú. 2. El sistema carga ventana. 3. La prueba ingresa datos. 4. El sistema guarda datos.
Valores de prueba	[Valores de las entradas utilizadas] Ejemplo Entrada1 = "A" Entrada2 = "B"
Resultados esperados	
	[Flujo de prueba y salidas del sistema] Ejemplo 01->02[V02]->03->04[V04]
V02	El sistema muestra una ventana con elementos de ingreso de Entrada1, Entrada2, Entrada n.
V04	El sistema muestra un mensaje de almacenamiento exitoso
Notas	[Notas adicionales sobre la realización de la prueba tales como precondiciones]

CASO DE ESTUDIO PRÁCTICO

Actividades pruebas unidad



Actividades pruebas integración



VALIDACIÓN

Metodología

- Comparación
- Evaluación expertos
- Resultados
- Discusión

Métricas

- Tiempo de desarrollo
- Loc

Comparativa Referencial

Marco de Trabajo	Apache struts 2.3.8 con hibernate 3.2.5 y jquery para funcionalidad asincrónica.
Tiempo de desarrollo	10 horas con 40 minutos = 10,67 horas
Caso Uso Referencia	CU2. Gestionar Administradores.
Novedades	Las librerías de ambos marcos de trabajo crean conflictos entre sí.

Proyectos:

- Web services
- GUI Struts 2.

VALIDACIÓN

Actividades Desarrolladas

Paso	Recurrencia	Tiempo
Mapeo	1 sola vez	10 min
Hibernate (capa de modelo)		
Hibernate	1 vez	2 min
Util (capa de modelo)		
Web service (capa de negocio)	Por cada interfaz y por cada funcionalidad	5 horas
Publicación del servicio web. (capa de negocio)	Por cada despliegue	5 minutos
Referencia Servicio	Cambios en los servicios que sean recién creados	1 minuto
Controlador	Por cada interfaz y funcionalidad	4 horas
Vista	n/a	1 hora

Capas

Capa	Tipo	LOC	Tiempo
Negocio	Desarrollada	163	5 horas
Controlador	Desarrollada	144	4 horas
Vista	Desarrollada	n/a	1 hora
Peticiones asincrónicas	Desarrollada	65	30 minutos
Validaciones	Desarrollada	43	10 minutos

Archivos de Configuración

Archivo	Descripción	LCF	TCF
web.xml	Configuración de filtros de servlets	3	1 min
struts.xml	Configuración de acciones hacia métodos de controlador	6	1 min cada línea

VALIDACIÓN

Marco de Trabajo	Propuesto.
Tiempo de desarrollo	3 horas con 20 minutos = 3,34 horas.
Caso Uso Referencia	CU2. Gestionar Administradores.
Novedades	Aplicado en el caso práctico documentado.

Actividades Desarrolladas

Paso	Recurrencia	Tiempo
Mapeo Relacional de objetos y parte de lógica de negocio. (capa de modelo, negocio)	1 sola vez	1 min
Implementación de lógica (capa de negocio)	Por cada interfaz y por cada funcionalidad	1 hora
Creación de controlador de vista	Por cada interfaz y funcionalidad	1 hora
Creación de vista	n/a	1 hora

Archivos de Configuración

Archivo	Descripción
config.xml	Configuración general de página inicial del marco de trabajo.
Widgets-js.xml	Configuración de elementos globales, se deja pre - configurada, solo sirve para expandir funcionalidades del marco

Capas

Capa	Tipo	LOC	T
Negocio	Modificada	52	1 hora
Controlador	Desarrollada	74	1 hora
Vista	Desarrollada	n/a	1 hora
Peticiones asincrónicas	Generada	n/a	0
Validaciones y complementarios	Desarrollada	36	20 minutos

VALIDACIÓN

Evaluación Expertos

$$n = \frac{Z^2 P Q N}{(N - 1) E^2 + Z^2 P Q}$$

Población: 150

n= Tamaño de muestra

Z= Valor Z curva normal (1.96)

P= Probabilidad de éxito (0.50)

Q= Probabilidad de fracaso (0.50)

N= Población (150)

E= Error muestral (0.05)

$$n = 108,0801 \sim 108$$

No.	Pregunta	Opción	Respuestas	Total	%
1	Arquitectura de capas.	Si	108	108	100
		No	0		0
2	Marco de trabajo modelo - vista - controlador.	Si	108	108	100
		No	0		0
3	Aplicaciones web de tiempo real.	No	80	108	74
		Si	28		26
4	Tecnología de tiempo real.	No	80	108	74
		Si	28		26

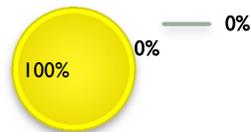
No.	Pregunta	Opción	Respuestas	Total	%
5	Qué marco de trabajo considera el mejor.	Struts 2	0	108	0
		Spring	0		0
		Symfony	0		0
		Zend	0		0
		Asp. MVC	80		74
		Prime Faces	0		0
		Ice Faces	0		0
		Rich Faces	0		0
		Otros	28		26
6	Marco de trabajo utilizado de preferencia soporta tecnología RTW.	No	94	108	87
		Si	14		13
7	Interés RTW	Si	108	108	100
		No	0		0

VALIDACIÓN

Sencillez

8. Considera intuitivo el flujo de pasos para la realización de interfaces asincrónicas y de tiempo real.

- a) Totalmente de acuerdo.
- b) Mayoritariamente de acuerdo.
- c) Parcialmente de acuerdo.
- d) En desacuerdo.



9. Considera comprensible el lenguaje de marcado para vistas.

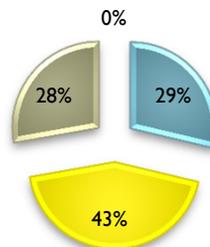
- a) Totalmente de acuerdo.
- b) Mayoritariamente de acuerdo.
- c) Parcialmente de acuerdo.
- d) En desacuerdo.



Claridad

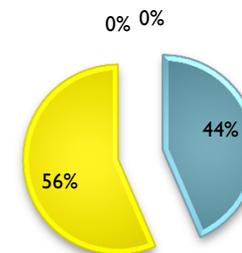
10. Considera clara la disposición de elementos del patrón modelo vista controlador con reglas precisas.

- a) Totalmente de acuerdo.
- b) Mayoritariamente de acuerdo.
- c) Parcialmente de acuerdo.
- d) En desacuerdo.



11. Considera adecuadas la cantidad de capas utilizadas en la constitución de una funcionalidad.

- a) Totalmente de acuerdo.
- b) Mayoritariamente de acuerdo.
- c) Parcialmente de acuerdo.
- d) En desacuerdo.

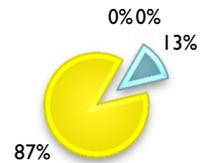


VALIDACIÓN

Delimitación

12. Considera la existencia de una restricción adecuada del estilo de creación de elementos funcionales en la aplicación del patrón modelo vista controlador.

- a) Totalmente de acuerdo.
- b) Mayoritariamente de acuerdo.
- c) Parcialmente de acuerdo.
- d) En desacuerdo.



Independencia

13. Considera conveniente la separación y reuso de los elementos del mapeo relacional de acceso a datos.

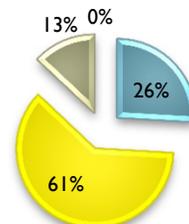
- a) Totalmente de acuerdo.
- b) Mayoritariamente de acuerdo.
- c) Parcialmente de acuerdo.
- d) En desacuerdo.



Modularidad

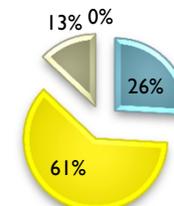
14. Considera conveniente la organización agrupada de elementos del patrón MVC propuesto.

- a) Totalmente de acuerdo.
- b) Mayoritariamente de acuerdo.
- c) Parcialmente de acuerdo.
- d) En desacuerdo.



15. Considera modular la organización de elementos en el esquema propuesto para construir las funcionalidades asincrónicas y de tiempo real.

- a) Totalmente de acuerdo.
- b) Mayoritariamente de acuerdo.
- c) Parcialmente de acuerdo.
- d) En desacuerdo.

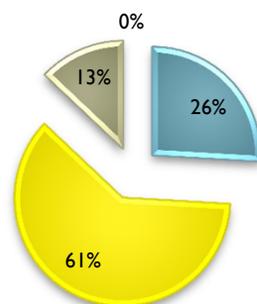


VALIDACIÓN

Expansibilidad

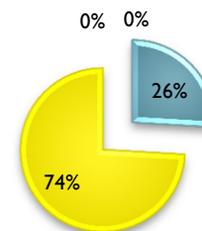
16. Considera escalables los patrones de desarrollo del marco de trabajo propuesto.

- a) Totalmente de acuerdo.
- b) Mayoritariamente de acuerdo.
- c) Parcialmente de acuerdo.
- d) En desacuerdo.



17. Considera que el montaje de funcionalidades asincrónicas y de tiempo real con la propuesta para que puedan ser implementados rápidamente con los patrones de desarrollo del marco de trabajo propuesto.

- a) Totalmente de acuerdo.
- b) Mayoritariamente de acuerdo.
- c) Parcialmente de acuerdo.
- d) En desacuerdo.



VALIDACIÓN

Discusión

Resultados
Métricas

Marco Trabajo
Propuesto

Marco Trabajo
Comparación

- Mejora velocidad Construcción.
- ORM y Negocio parcialmente Generado.
- Enfoque actividades negocio.
- Asíncrono nativamente.

- Sacrifica Tiempo.
- ORM Generado – Negocio Desarrollarse
- Ajustar arquitectura.
- Adaptar funcionalidad asíncrona.

Resultados
Criterio
Expertos

- Desarrollo por Capas.
- MVC.
- Interés Tiempo Real.

Características y
patrones desarrollo

- Sencillez.
- Claridad.
- Delimitación.
- Independencia.
- Modularidad.
- Expansibilidad.



Acceptable

CONCLUSIONES

- Marco de Trabajo
 - Patrones.
 - Prácticas.
 - Arquitectura.
 - Propuesto acelera - RTW.
- Tiempo Real
 - Suscripción.
 - Disparo.
 - Arquitectura.
 - Integración MT, Independencia,.
 - Enlace.
- Capacidades
 - Características Equipos.
- RAD
 - Marco de Trabajo.
 - RTW.
 - Micrositios.
 - Aplicable TICs.
- Tecnología Ajax Inverso
 - Aprovechamiento.
 - Proyectos Micrositios.
 - Innovación.

- RTW Micrositios
 - Innovadores.
 - Interactivos.
 - Red Social.
- Modelo Información
 - Mejora.
 - Experiencia.
 - Comunidad.

RECOMENDACIONES

Micrositios

- Tics.
- Marco de Trabajo.
- Modelo tecnológico.

Futuras Investigaciones

- Expansión.
- Otros Dispositivos.
- Aplicaciones Ubicuas.
- Interés.

Campo Nuevo

- Productos.
- Mercado Nacional.
- Propuesta.

Diseño Marco de Trabajo

- Existentes.
- Características.
- Modelo Desarrollo.

Plantear

- Prototipo.
- Aplicación.
- Validación.

Apoyo

- Lugar Aplicación.
- Agilitar.
- Desarrollo.



MUCHAS GRACIAS