

Battle Assistant

Àrea TFG: Java EE

Autor: Luis Manuel Martín Guerra

Consultor: Albert Grau Perisé

10 de Enero 2018

Contenidos

- ▶ Introducción.
 - ▶ Contexto.
 - ▶ Objetivos.
 - ▶ Planificación.
 - ▶ Enfoque y Metodología.
- ▶ Análisis y Diseño.
 - ▶ Requisitos.
 - ▶ Especificación.
 - ▶ Prototipos.
 - ▶ Arquitectura.
- ▶ Implementación.
 - ▶ Capas de la aplicación.
 - ▶ Análisis.
- ▶ Demostración.
- ▶ Conclusiones.

Introducción

- ▶ Contexto.
- ▶ Objetivos.
- ▶ Planificación.
- ▶ Enfoque y Metodología.

Contexto

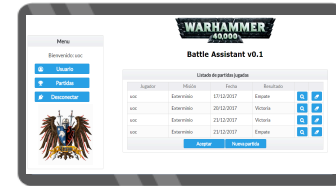
En el siniestro futuro del cuadragésimo primer milenio, solo hay...

- ▶ Muchos manuales.
- ▶ Tiempo perdido.
- ▶ Interrupciones.



Objetivos

- ▶ Crear una aplicación web que simplifique y agilice las partidas.
- ▶ Poner en práctica los conocimientos adquiridos a lo largo de la carrera.
- ▶ Profundizar en las tecnologías de Java EE.
- ▶ Desarrollar un producto software desde cero.



Planificación

El proyecto se ha dividido en las siguientes fases:

- ▶ Creación del plan de trabajo - PAC1 (20/09/2017) - (02/10/2017)
- ▶ Análisis y Diseño - PAC2 (03/10/2017) - (06-11-2017)
- ▶ Implementación - PAC3 (04/11/2017) - (21/12/2017)
- ▶ Documentación - Memoria y Presentación (22/12/2017) - (10/01/2017)

Enfoque y metodología

Uso de sprints de 5 etapas:



Análisis y Diseño

- ▶ Requisitos.
- ▶ Especificación.
- ▶ Esquemas y Prototipos.
- ▶ Arquitectura.

Requisitos

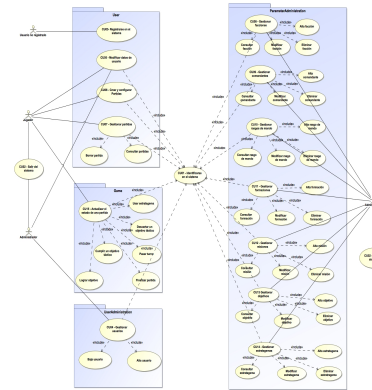
- ▶ Características principales.
- ▶ Requisitos funcionales.
- ▶ Requisitos no funcionales.

Especificación

- ▶ Listado de casos de uso.
- ▶ Diagrama de casos de uso.
- ▶ Fichas de caso de uso.

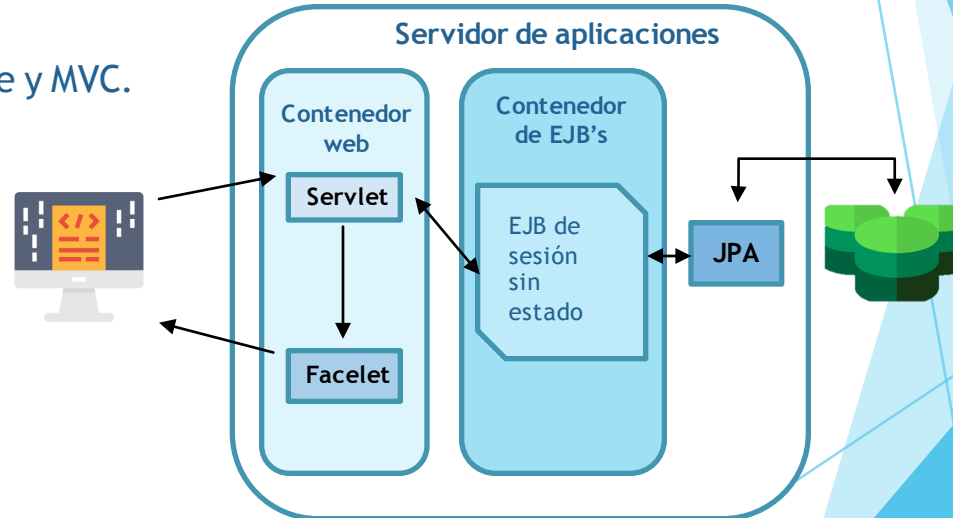
Caso de uso: Actualizar el estado de la partida	
Identificador:	CU15
Actor principal:	Jugador
Ámbito:	Sistema
Pre-condición:	El Jugador debe estar identificado en el sistema (CU01) y haber creado una partida
Descripción:	El Jugador va indicando al sistema los diferentes eventos que se producen en la partida y este actualiza los datos de las variables que monitoriza.
Escenario Principal:	<ol style="list-style-type: none"> El Jugador va indicando al sistema las diferentes acciones que van realizando los jugadores a lo largo de una partida: <ol style="list-style-type: none"> Usar una estrategia (CU16) Lograr un objetivo (CU17) Cumplir un objetivo táctico (CU18) Pasar turno (CU19) Finalizar la partida (CU20) El sistema va actualizando los datos de la partida

Caso de uso: Gestionar partidas	
Identificador:	CU07
Actor principal:	Jugador
Ámbito:	Sistema
Pre-condición:	El Jugador debe estar identificado en el sistema (CU01).
Descripción:	El Jugador consulta los detalles de una partida que ha jugado anteriormente.
Escenario Principal:	<ol style="list-style-type: none"> El Jugador indica al sistema que quiere ver las partidas que ha jugado. El sistema muestra un listado con las partidas que ha jugado el Jugador. El Jugador selecciona una partida del listado. El sistema muestra la información de la partida jugada. El Usuario pulsa el botón de aceptar. El sistema vuelve a la lista de partidas del Jugador.
Escenario Alternativo:	<ol style="list-style-type: none"> El usuario indica al sistema que quiere borrar una partida. <ol style="list-style-type: none"> El sistema muestra un mensaje de confirmación. El caso de uso vuelve al punto 2.



Arquitectura

- ▶ Arquitectura en 3 capas.
- ▶ Aplicación de los patrones Facade y MVC.
- ▶ Uso de los estándares Java EE.
 - ▶ Persistencia: JPA.
 - ▶ Negocio: EJB.
 - ▶ Presentación: JavaServer Faces + Primefaces.



Implementación

- ▶ Herramientas.
- ▶ Capas de la aplicación.
- ▶ Demostración.
- ▶ Análisis y futuras mejoras.

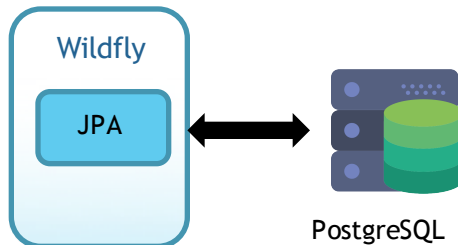
Herramientas software

- ▶ Entorno de desarrollo Eclipse (Oxygen 4.7.1).
- ▶ Sistema gestor de bases de datos PostgreSQL (9.6.5-1).
- ▶ Conector JDBC PostgreSQL.
- ▶ Servidor de aplicaciones Wildfly (10.1).



Capa de Integración

- ▶ Uso de entidades JPA.
- ▶ Agrupadas por paquetes.
- ▶ Estrategia de tablas intermedias.

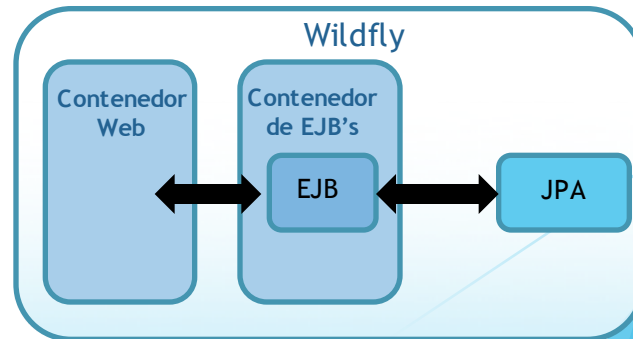


- ▼ game.jpa
 - > ArmyJPA.java
 - > GameJPA.java
- ▼ parameterAdministration.jpa
 - > CommanderJPA.java
 - > DeploymentJPA.java
 - > FactionJPA.java
 - > FactionObjectiveJPA.java
 - > FactionStratagemJPA.java
 - > FormationJPA.java
 - > MissionJPA.java
 - > MissionStratagemJPA.java
 - > ObjectiveJPA.java
 - > RuleJPA.java
 - > TraitJPA.java
- ▼ userAdministration.jpa
 - > UserJPA.java

Capa de Negocio

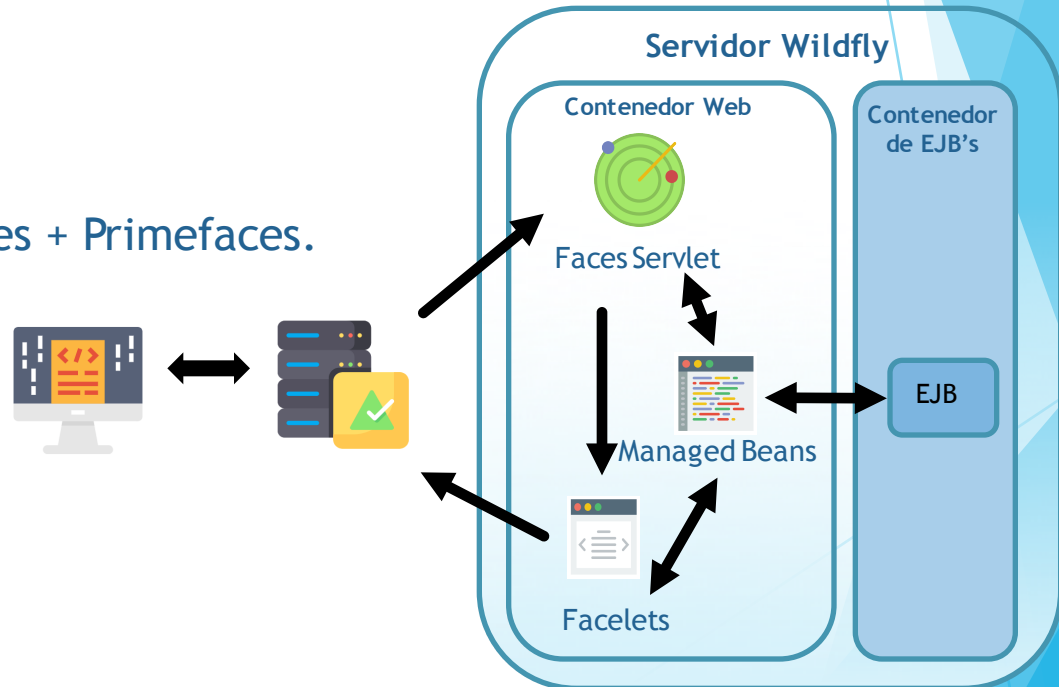
- ▶ Uso del patrón Facade (Fachada).
- ▶ Implementación con EJB's.
- ▶ Clase auxiliar IndexCalculator.

- ▼ game.ejb
 - > GameFacadeBean.java
 - > GameFacadeRemote.java
- ▼ parameterAdministration.ejb
 - > ParameterFacadeBean.java
 - > ParameterFacadeRemote.java
- ▼ userAdministration.ejb
 - > UserFacadeBean.java
 - > UserFacadeRemote.java



Capa de Presentación

- ▶ Uso del patrón MVC.
- ▶ Uso de JavaServer Faces + Primefaces.
- ▶ Uso de @WebFilter.
- ▶ Uso de plantillas.



Demostración

Vamos a ver el funcionamiento de la aplicación desarrollada.



Battle Assistant v0.1

Acceso al sistema

Usuario: *

Contraseña: *

¿No estás registrado? [Regístrate aquí](#)

Análisis y futuras mejoras

- ▶ Identificación: implementar acceso mediante JAAS.
- ▶ Gestión de usuarios: implementar registro asíncrono.
- ▶ Usuario: generar y guardar ejércitos.
- ▶ Parámetros del sistema: subida de imágenes al servidor.

Conclusiones

- ▶ Oportunidad de poner en práctica conocimientos adquiridos.
- ▶ Primera toma de contacto con el mundo “real”.
- ▶ Expectativas altas, resultados dispares.
- ▶ Mejora de habilidades y nuevos conocimientos.
- ▶ Aplicación web base para desarrollar nuevos módulos.
- ▶ Experiencia muy positiva.

*“Me lo contaron y lo olvidé;
Lo ví y lo entendí;
Lo hice y lo aprendí”.*