

Desarrollo de Aplicaciones Distribuidas

Rafael Gálvez-Cañero, Andreas Gerstmayr

Iteración 1 - 15 de Febrero de 2015

Índice general

1. Datos generales	2
1.1. Miembros del grupo	2
1.2. Descripción del sistema	2
1.2.1. Funcionalidad observable	2
1.2.2. Servicios ofrecidos	2
1.2.3. Servicios demandados	3
1.3. Direcciones de descarga y planificación	3
1.4. Seguimiento	3
2. Iteración 2	4
2.1. Modelo de Análisis	4
2.2. Arquitectura del sistema	4
3. Iteración 3	5
3.1. Planificación	5
3.2. Diseño	5
3.2.1. Diagrama de clases UML	5
3.2.2. Memorandos técnicos	7
3.3. Información adicional	7
4. Iteración 4	8
4.1. Planificación	8
4.2. Diseño	8
4.2.1. Diagrama de clases UML	8
4.2.2. Documentos de asignación de responsabilidades	8
4.2.3. Memorandos técnicos	8
4.3. Información adicional	8

Índice de figuras

2.1. Modelo de análisis del sistema	4
2.2. Modelo de análisis del sistema	4
3.1. Diagrama UML de diseño para la iteración 3	5
4.1. Diagrama UML de diseño para la iteración 4	8

Índice de cuadros

1.1. Miembros del grupo	2
1.2. Datos generales del trabajo en grupo	3
1.3. Tabla de seguimiento	3

Capítulo 1

Datos generales

1.1. Miembros del grupo

Apellidos	Nombre	Correo-e	Grupo
Gálvez-Cañero	Rafael	rafgalgal1@alum.us.es	18
Gerstmayr	Andreas	andreas.gerstmayr@gmail.com	18

Cuadro 1.1: Miembros del grupo

1.2. Descripción del sistema

- **Tipo de sistema distribuido:**
- **Nombre del proyecto:** Plataforma de juegos, Game Register
- **Breve descripción:**

1.2.1. Funcionalidad observable

- Registrar el inicio y el término de las todas de juego sesiones.
- Visualizar la historia de juegos
- Visualizar qué jugadores juegan en este momento

1.2.2. Servicios ofrecidos

- Servicio XYZ: breve descripción y tecnologías a través de la que se ofrece. Fecha aproximada de disponibilidad
- Servicios ZYX: idem.

1.2.3. Servicios demandados

- Servicio X: breve descripción. Fecha aproximada a partir de la que se necesitará (si se conoce).
- Servicio Y: idem.

1.3. Direcciones de descarga y planificación

Código fuente	https://repositorio.informatica.us.es/svn/lq3vqrtzfznh2nx9yhp
Planificación temporal	
Iteración 1	15/02/2015
Iteración 2	01/03/2015
Iteración 3	15/03/2015
Iteración 4	05/04/2015
Iteración 5	19/04/2015
Iteración 6	10/05/2015
Iteración 7	24/05/2015
Entrega Final	07/06/2015

Cuadro 1.2: Datos generales del trabajo en grupo

1.4. Seguimiento

Estudiante	Iteración								Final	Total	Pond.
	1	2	3	4	5	6	7	8			
Rafael Gálvez-Cañero	5	5	5	5	5	5	5	5	5	40	1
Andreas Gerstmayr	5	5	5	5	5	5	5	5	5	40	1
Total	30	30	30	30	30	30	30	30	30		

Cuadro 1.3: Tabla de seguimiento

Capítulo 2

Iteración 2

2.1. Modelo de Análisis

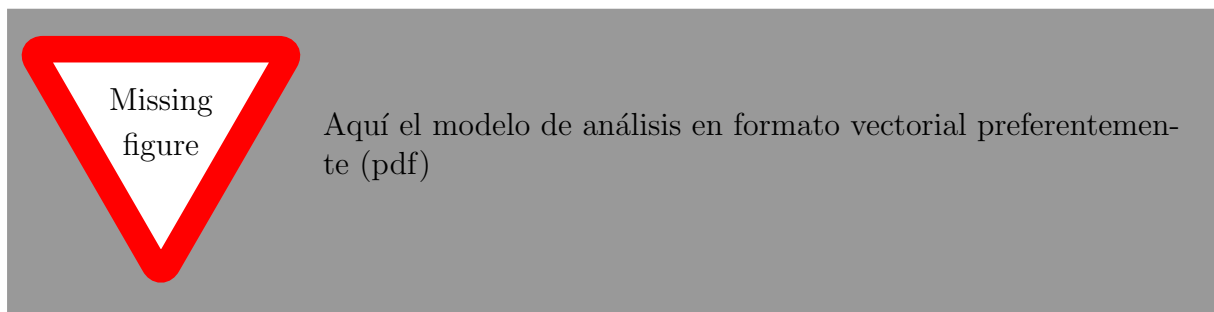


Figura 2.1: Modelo de análisis del sistema

2.2. Arquitectura del sistema

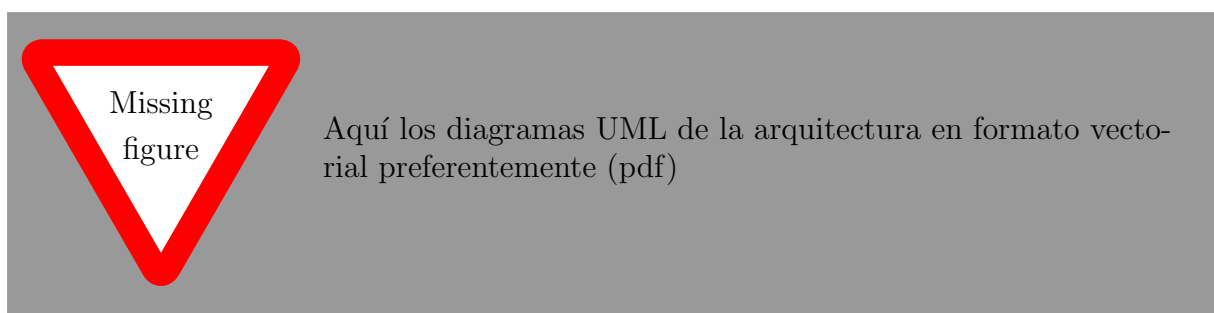


Figura 2.2: Modelo de análisis del sistema

Capítulo 3

Iteración 3

3.1. Planificación

Tareas realizadas en esta iteración.

Tareas para la siguiente iteración.

Modificación si procede de la planificación.

3.2. Diseño

3.2.1. Diagrama de clases UML

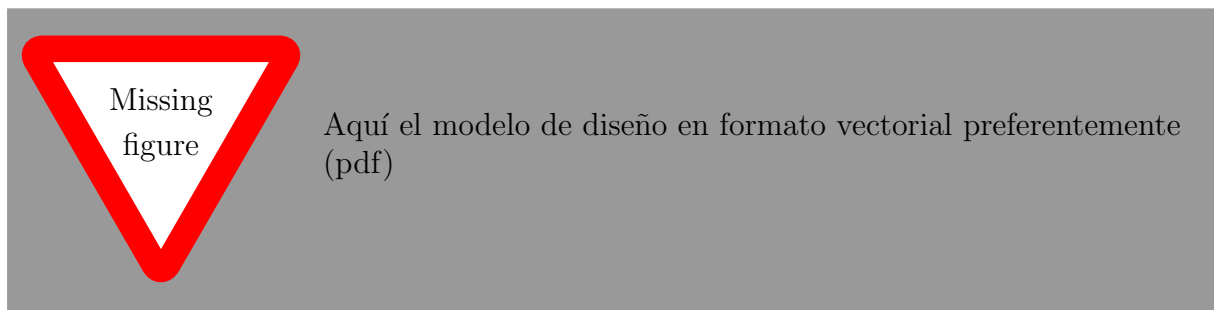
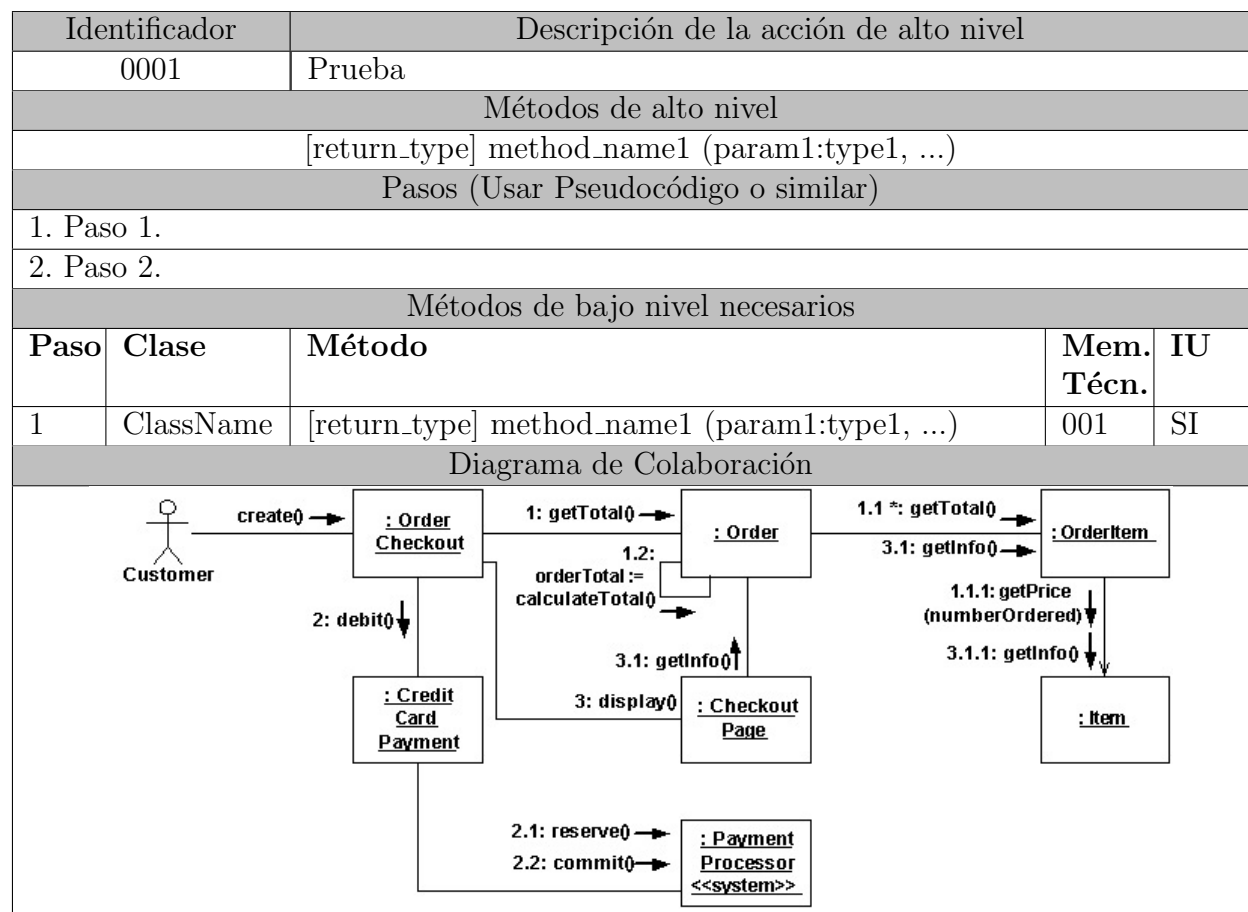


Figura 3.1: Diagrama UML de diseño para la iteración 3



Identificador	Descripción de la acción de alto nivel			
alvotermar02	Grubber			
Métodos de alto nivel				
[return_type] grubber (param1:type1, ...)				
Pasos (Usar Pseudocódigo o similar)				
1. Lanzar 2 dados				
2. Compara resultado de los dados con kicking del open-side				
2.1. Si valor dados es menor o igual a kicking, avanza 10m				
3.1. Si no hay defensa y el golpeo es exitoso, el pateador retiene la posesión del balón				
3.2. Si hay defensa y el golpe es exitoso, el atacante tira un dado y suma su valor al de speed y strength y el defensor lanza 2 dados y lo suma al valor de speed y strength de su jugador, el vencedor será aquel que tenga más puntos, si es igual, la posesión es del defensor				
4.1. Si no es exitoso y hay defensa el balón pasa a posesión del defensor				
4.2. Si no es exitoso y no hay defensa de lanza un line-out				
Métodos de bajo nivel necesarios				
Paso	Clase	Método	Mem. Técn.	IU
1	Dice	[Integer] throwDice ()	001	SI
2	ClassName	[Int] compareKickingToDice (kicking:Integer, dice: Integer)	001	SI
2.1	ClassName	[Integer] setLine (line:Integer)	001	SI
4.2	ClassName	[Integer] lineOut ()	001	SI

3.2.2. Memorandos técnicos

Identificador: Formato libre	
Asunto:	¿Cuál es el problema?
Resumen de la solución:	¿Cuál es la solución propuesta?
Factores causantes:	Descripción pormenorizada del problema
Solución:	Descripción pormenorizada de la solución propuesta
Motivación:	Por qué propone esta solución?
Cuestiones sin resolver:	Factores a tener en cuenta en la solución cuya dimensión se reconoce.
Alternativas consideradas:	Otras soluciones consideradas y la razón por la que se excluyeron.

3.3. Información adicional

La que proceda como detalles de implementación y/o pruebas

Capítulo 4

Iteración 4

4.1. Planificación

4.2. Diseño

4.2.1. Diagrama de clases UML

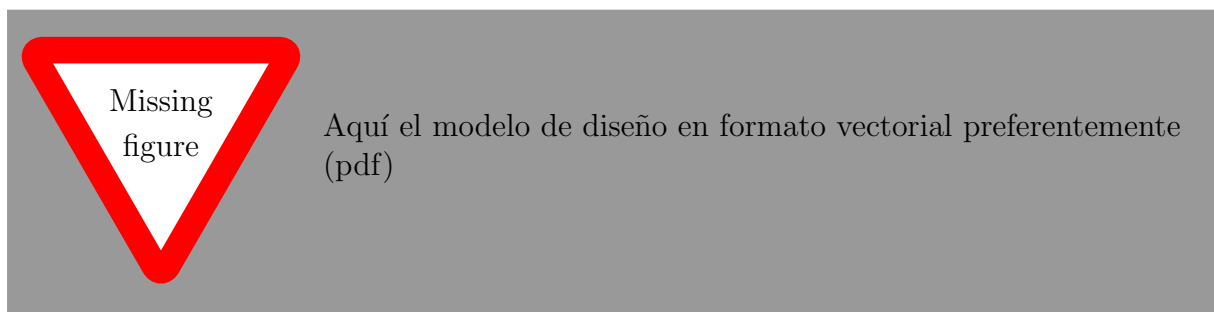


Figura 4.1: Diagrama UML de diseño para la iteración 4

4.2.2. Documentos de asignación de responsabilidades

4.2.3. Memorandos técnicos

4.3. Información adicional