

Prácticas de Software de Sistema de Recompensas

Diseño de Sistemas Software

Curso: 2018/2019, Versión 1.0.0

Índice

1. Introducción	3
1.1. Concepto Inicial	3
1.2. Aspectos técnicos	3
1.3. Documentación	4
1.4. Modelo de trabajo colaborativo	4
1.5. Evaluación de la práctica	4
2. Requisitos funcionales	5
2.1. Modelo de casos de uso	5
2.2. Definición de casos de uso	6
3. Programa principal	8
3.1. Menú principal	9
3.2. CR jugador	9
3.2.1. Añadir nuevo jugador	9
3.2.2. Consultar datos de un jugador	10
3.3. CRUD videojuego	10
3.3.1. Añadir nuevo videojuego	11
3.3.2. Editar datos de un videojuego	11
3.3.3. Consultar datos de un videojuego	11
3.3.4. Eliminar videojuego	12
3.4. CRUD recompensa	12
3.4.1. Añadir nueva recompensa	12
3.4.2. Editar datos de una recompensa	13
3.4.3. Consultar datos de una recompensa	13
3.4.4. Eliminar recompensa	14
3.5. CR DLC	14
3.5.1. Añadir nuevo DLC	14
3.5.2. Consultar datos de un DLC	15
3.6. Notificar nuevo jugador de un videojuego	15
3.7. Notificar obtención de recompensa de un videojuego	15
3.8. Mostrar estadísticas	16
3.8.1. Consultar jugadores por videojuego	16
3.8.2. Consultar estadísticas de las recompensas por videojuego	16

3.8.3. Consultar jugadores que han completado cada videojuego	17
4. Pruebas del sistema	17
5. Entregables	18
5.1. Primera entrega: Diseño preliminar	18
5.2. Segunda entrega: Diseño con Principios	19
5.3. Tercera entrega: Implementación con memoria	19
5.4. Cuarta entrega: Implementación persistente	19
5.5. Quinta entrega: Pruebas unitarias	19
5.6. Sexta entrega: Final	20
6. Evaluación	20

1. Introducción

Este documento describe la práctica que se realizará en la asignatura de **Diseño de Sistemas Software (DSS)** para el curso 2018/2019.

El objetivo de la práctica es la realización de un sistema software incluyendo tanto su diseño, como su implementación, a partir de la especificación aportada en el presente documento.

Para posibilitar un correcto seguimiento, y poder aprender mejor durante el proceso, la entrega de la práctica se realizará en distintas entregas incrementales de funcionalidad (hitos).

1.1. Concepto Inicial

El objetivo de la práctica es el diseño e implementación de un sistema software que gestione el sistema de recompensas de los jugadores de una determinada plataforma de videojuegos.

Este sistema permitirá:

1. Añadir nuevos jugadores.
2. Añadir nuevos videojuegos con su respectiva colección de recompensas.
3. Actualizar las recompensas de un determinado juego para añadir las asociados a un nuevo contenido descargable (DLC) del juego.
4. Cuando un jugador utilice un nuevo videojuego se debe añadir a la lista de videojuegos de este jugador.
5. Cuando un jugador consiga una determinada recompensa se debe registrar en la lista de usuarios que la han conseguido.
6. Mostrar los jugadores que poseen un determinado videojuego.
7. Consultar de forma global las recompensas de un determinado videojuego visualizando el porcentaje de obtención por cada uno de ellos, teniendo en cuenta los usuarios que poseen el juego.

1.2. Aspectos técnicos

Para la implementación de la práctica se debe hacer uso de las siguientes herramientas:

- Lenguaje de programación: Java. **JavaScript (Angular)**
- Entorno de desarrollo: Eclipse IDE.
- Pruebas unitarias: Junit. **Jasmine**
- Repositorio git: GitHub.
- Herramienta de integración continua: Travis CI. **Gulp**
- Almacenamiento: MySQL.

1.3. Documentación

Una vez planteada la especificación de la aplicación, es responsabilidad de cada grupo realizar el diseño que considere más conveniente. Durante la asignatura estudiaremos criterios para mejorar el diseño.

Dado que el diseño puede modificarse durante el proceso, cada entrega implica la presentación del documento de diseño. Por motivos de facilidad en la documentación, y para facilitar el trabajo colaborativo, dicho documento se realizará como documentación del repositorio de GitHub donde se vaya a alojar el código fuente.

Dicha documentación debe contener los siguientes apartados:

- Explicación general del sistema.
- Nombres de los miembros del grupo.
- Reparto de tareas entre los distintos miembros del grupo.
- Diagrama de clases (UML) de usuario, indicando tanto las clases como sus relaciones y principales métodos públicos (no en las entregas iniciales).
- Patrones de diseño aplicados (el concepto se verá en clase).
- Justificación de las decisiones de diseño (este apartado es importante).
- Ejemplos de uso de los distintos casos de uso.

1.4. Modelo de trabajo colaborativo

Uno de los aspectos claves para la práctica es el trabajo en equipo, y como tal se ha definido la dificultad de la práctica. Para la correcta realización de la práctica es necesario la colaboración de los distintos miembros del equipo.

Dado que la primera entrega es el proceso de diseño general, todos deberán de participar activamente. Una vez pasada esa fase, para poder realizar una correcta coordinación, cada miembro del equipo deberá de definir sus tareas. Debido a su importancia, se indicarán el reparto de tareas correspondiente, que deberá de ser aprobado por el profesor. Como el resto de documentación, podrá ser actualizada.

La calificación de cada estudiante dependerá de la calificación global de las prácticas, aunque con una especial valoración en función a las tareas de las que haya sido responsable.

Para favorecer el trabajo en equipo la documentación y el código fuente deben estar disponibles en el repositorio de GitHub.

1.5. Evaluación de la práctica

Para realizar la evaluación se definirá una *rúbrica* u hoja de cálculo con los criterios indicados para evaluar la práctica. Dicha hoja de cálculo establece una serie de criterios, y una valoración numérica en función del grado de cumplimiento de los mismos (también indicados en el documento).

Este documento se presentará de forma previa a las entregas y permitirá servir como plantilla para que cada grupo pueda evaluar su propia práctica. La entrega final deberá obligatoriamente de ir acompañada

de una auto-evaluación de la misma, utilizando esta plantilla. Evidentemente, la calificación final de la entrega estará en función de la valoración del profesor, no del propio grupo, pero se presenta la plantilla para que quede claro el criterio de autoevaluación.

Dado que se realizarán distintas entregas, y se seguirá un proceso incremental, en una entrega se podrán arreglar los aspectos deficitarios de la entrega anterior, y arreglando su puntuación correspondiente, pero con una penalización de un 20 %. Es decir, se deberán arreglar defectos de entregas anteriores, y se valorará en la calificación, pero sin llegar a obtener la máxima calificación posible, para favorecer a aquellos que cumplieron adecuadamente cada criterio en entregas tempranas.

2. Requisitos funcionales

2.1. Modelo de casos de uso

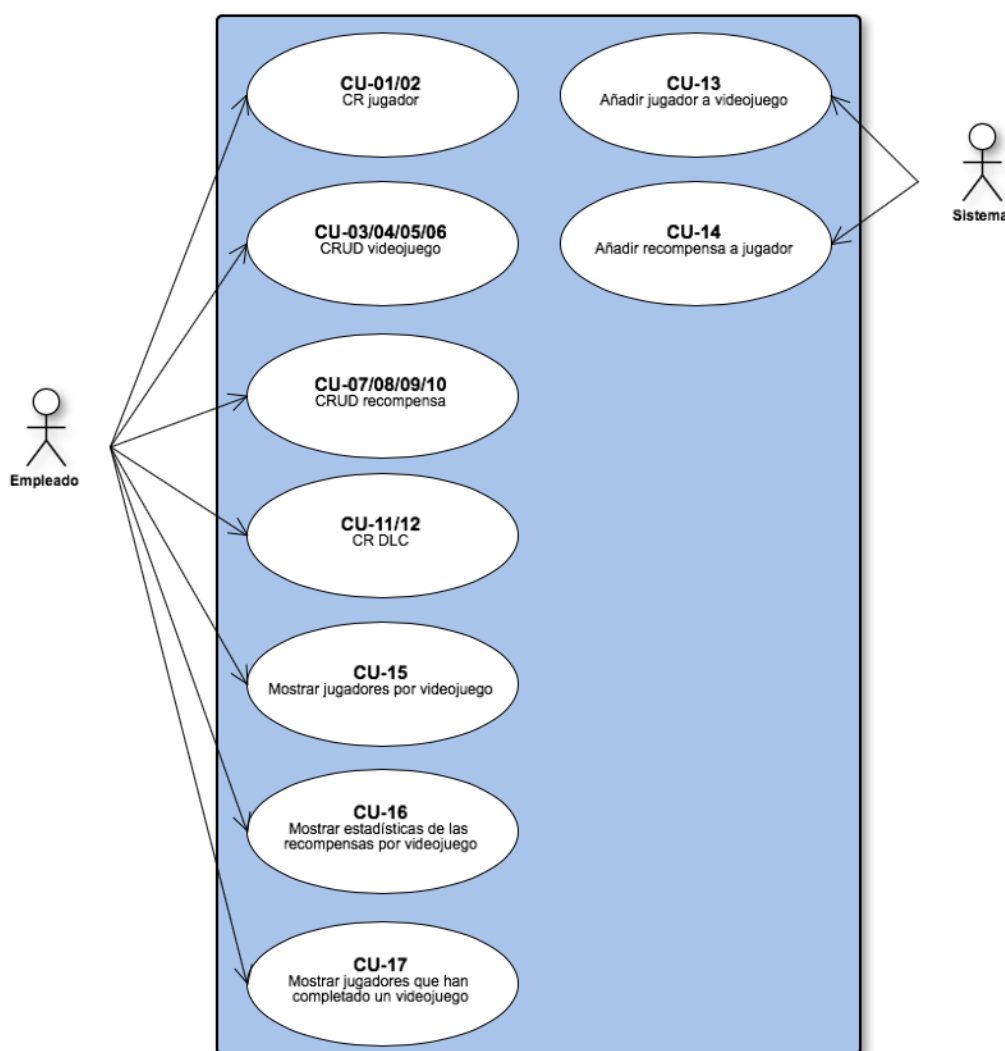


Figura 1: Modelo de casos de uso del sistema de recompensas

2.2. Definición de casos de uso

A continuación, se definen los casos de uso del sistema:

C01: createJugador Un jugador se registra en el sistema. Para facilitar la práctica y no tener múltiples usuarios en la aplicación supondremos que el empleado da de alta al jugador en el sistema.

1. El empleado introduce nombre y apellidos.
2. El empleado introduce el nickname.
3. El empleado introduce correo electrónico del usuario.
4. El empleado introduce la fecha de nacimiento del usuario.
5. El empleado guarda de forma persistente el cambio.

C02: readJugador En este caso el trabajador de la empresa podrá consultar los datos asociados a un determinado jugador.

1. El empleado selecciona el nickname del jugador a visualizar.
2. El sistema muestra el listado de los campos del jugador seleccionado.

C03: createVideojuego En este caso el trabajador de la empresa deberá poder añadir un nuevo videojuego a la lista de los existentes.

1. El empleado introduce el título del videojuego.
2. El empleado introduce la descripción del videojuego.
3. El empleado elige el género del videojuego de entre los existentes en el sistema: Acción, Adultos, Aventura, Arcade, Agilidad mental, Casual, Educativo, Familia, Lucha, Fitness, Horror, Música y ritmo, Grupo, Puzzle, Carreras, Juegos de Rol, Shooter, Simulador, Deportes, Estrategia y Único. Un juego puede tener varios géneros asociados.
4. El empleado elige el dispositivo asociado al videojuego. Ejemplos: PS4, Xbox One, Ps Vita, Xbox 360, PC, etc.
5. Se guarda de forma persistente el cambio.

C04: readVideojuego En este caso el trabajador de la empresa deberá poder listar los detalles de un determinado videojuego.

1. El empleado selecciona el título del videojuego a visualizar.
2. El sistema muestra el listado de los campos del videojuego seleccionado.

C05: updateVideojuego En este caso el trabajador de la empresa deberá poder actualizar un videojuego existente en la lista.

1. El empleado selecciona el título del videojuego a actualizar.
2. El empleado selecciona el nombre del campo a actualizar.

3. El empleado actualiza el valor del campo seleccionado.
4. Se guarda de forma persistente el cambio.

C06: deleteVideojuego En este caso el trabajador de la empresa deberá poder eliminar un videojuego existente en la lista.

1. El empleado selecciona el título del videojuego a eliminar.
2. El sistema pide confirmación de la eliminación de dicho título del catálogo.
3. El empleado confirma la eliminación.
4. El sistema elimina el videojuego y todas sus recompensas y DLC asociados.

C07: createRecompensa En este caso el trabajador de la empresa deberá poder añadir una nueva recompensa a la lista de los existentes.

1. El empleado introduce el título de la recompensa.
2. El empleado introduce la dificultad de la recompensa.
3. El empleado asigna la recompensa a la lista de recompensas del videojuego al que pertenece (o a la lista del DLC al que pertenece).
4. Se guarda de forma persistente el cambio.

C08: readRecompensa En este caso el trabajador de la empresa deberá poder listar los detalles de una determinada recompensa.

1. El empleado selecciona el título de la recompensa a visualizar.
2. El sistema muestra el listado de los campos de la recompensa seleccionada.

C09: updateRecompensa En este caso el trabajador de la empresa deberá poder actualizar una recompensa existente en la lista.

1. El empleado selecciona el título de la recompensa a actualizar.
2. El empleado selecciona el nombre del campo a actualizar.
3. El empleado actualiza el valor del campo seleccionado.
4. Se guarda de forma persistente el cambio.

C10: deleteRecompensa En este caso el trabajador de la empresa deberá poder eliminar una recompensa existente en la lista.

1. El empleado selecciona el título de la recompensa a eliminar.
2. El sistema pide confirmación de la eliminación de dicha recompensa del catálogo.
3. El empleado confirma la eliminación.
4. El sistema elimina la recompensa.
5. Se guarda de forma persistente el cambio.

C11: createDLC En este caso el trabajador de la empresa deberá poder añadir un nuevo contenido descargable a la lista de los existentes.

1. El empleado introduce el título del contenido descargable.
2. El empleado introduce la descripción del contenido descargable.
3. El empleado asigna el DLC a la lista de contenido descargable del videojuego al que pertenece.
4. Se guarda de forma persistente el cambio.

C12: readDLC En este caso el trabajador de la empresa deberá poder listar los detalles de un determinado contenido descargable.

1. El empleado selecciona el título del contenido descargable a visualizar.
2. El sistema muestra el listado de los campos del contenido descargable seleccionado.

C13: addJugador Cuando un usuario de la plataforma introduce un videojuego en su consola por primera vez el sistema debe registrarlo y actualizar las estadísticas globales de la plataforma.

1. El sistema añade el jugador a la lista de jugadores del videojuego y además, añade el videojuego a la lista de videojuegos del usuario.
2. Se guarda de forma persistente el cambio.

C14: addRecompensa Cuando un usuario de la plataforma consigue una determinada recompensa el sistema debe registrarlo y actualizar las estadísticas globales de la plataforma.

1. El sistema añade la recompensa a la lista de recompensas conseguidas para el videojuego concreto a dicho jugador.
2. Se guarda de forma persistente el cambio.

C15: showJugadoresPorVideojuego En este caso el trabajador debe poder visualizar los jugadores totales por cada videojuego.

1. El empleado selecciona el título del videojuego del que quiere observar las estadísticas.
2. El sistema muestra el listado de jugadores que tienen dicho videojuego.

C16: showRecompensasConseguidas En este caso el trabajador debe poder visualizar los jugadores totales que han conseguido cada una de las recompensas de un videojuego.

1. El empleado selecciona el título del videojuego del que quiere observar las estadísticas.
2. El sistema lista las recompensas del videojuego junto al porcentaje de usuarios que han conseguido cada una de ellas.

C17: showJuegosCompletados En este caso el trabajador debe poder visualizar los jugadores totales que han conseguido conseguir todas las recompensas para cada videojuego.

1. El empleado selecciona el título del videojuego del que quiere observar las estadísticas.
2. El sistema lista los jugadores que han conseguido la totalidad de recompensas para el videojuego seleccionado.

3. Programa principal

En esta sección se presentan unos ejemplos del contenido de las pantallas del sistema.

3.1. Menú principal

Menú principal

Acciones disponibles:

1. CR jugador
2. CRUD videojuego
3. CRUD recompensa
4. CR DLC
5. Notificar nuevo jugador de un videojuego
6. Notificar obtención de recompensa de un videojuego
7. Mostrar estadísticas

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.2. CR jugador

CR jugador

Acciones disponibles:

1. Añadir nuevo jugador
2. Consultar datos de un jugador

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.2.1. Añadir nuevo jugador

Añadir nuevo jugador

Acciones disponibles:

1. Añadir nombre y apellidos
2. Añadir nickname
3. Añadir correo electrónico
4. Añadir fecha de nacimiento

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.2.2. Consultar datos de un jugador

Datos: NickName

1. Nombre y apellidos: Nombre Apellidos
2. Nickname: Nickname
3. Correo electrónico: Email
4. Fecha de nacimiento: Fecha

R. Volver

Introduzca una opción:

3.3. CRUD videojuego

CRUD videojuego

Acciones disponibles:

1. Añadir nuevo videojuego
2. Editar datos de un videojuego
3. Consultar datos de un videojuego
4. Eliminar videojuego

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.3.1. Añadir nuevo videojuego

Añadir nuevo videojuego

Acciones disponibles:

1. Añadir título
2. Añadir descripción
3. Seleccionar género
4. Seleccionar dispositivo

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.3.2. Editar datos de un videojuego

Datos: Videojuego N

Acciones disponibles:

1. Editar título
2. Editar descripción
3. Editar género
4. Editar dispositivo

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.3.3. Consultar datos de un videojuego

Datos: Videojuego N

1. Título: Título
2. Descripción: Descripción
3. Género: Género
4. Dispositivo: Dispositivo

R. Volver

Introduzca una opción:

3.3.4. Eliminar videojuego

Datos: Videojuego N

¿Seguro desea eliminar el videojuego?, se borrarán todos las recompensas asociados.

Y. Sí

N. No

Introduzca una opción:

3.4. CRUD recompensa

CRUD recompensa

Acciones disponibles:

1. Añadir nueva recompensa
2. Editar datos de una recompensa
3. Consultar datos de una recompensa
4. Eliminar recompensa

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.4.1. Añadir nueva recompensa

Añadir nueva recompensa

Acciones disponibles:

1. Añadir título
2. Añadir descripción
3. Seleccionar dificultad
4. Seleccionar videojuego

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.4.2. Editar datos de una recompensa

Datos: Recompensa N

Acciones disponibles:

1. Editar título
2. Editar descripción
3. Editar dificultad
4. Editar videojuego

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.4.3. Consultar datos de una recompensa

Datos: Recompensa N

1. Título: Título
2. Descripción: Descripción
3. Dificultad: Dificultad
4. Videojuego: Videojuego

R. Volver

Introduzca una opción:

3.4.4. Eliminar recompensa

Datos: Recompensa N

¿Seguro desea eliminar la recompensa?.

Y. Sí

N. No

Introduzca una opción:

3.5. CR DLC

CR DLC

Acciones disponibles:

1. Añadir nuevo DLC
3. Consultar datos de un DLC

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.5.1. Añadir nuevo DLC

Añadir nuevo DLC

Acciones disponibles:

1. Añadir título
2. Añadir descripción

3. Seleccionar videojuego

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.5.2. Consultar datos de un DLC

Datos: DLC N

1. Título: Título
2. Descripción: Descripción
3. Videojuego: Videojuego

R. Volver

Introduzca una opción:

3.6. Notificar nuevo jugador de un videojuego

Añadir nuevo juego a jugador N

1. Seleccionar nuevo juego

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.7. Notificar obtención de recompensa de un videojuego

Añadir nueva recompensa del videojuego N al jugador J

1. Seleccionar nueva recompensa

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.8. Mostrar estadísticas

```
-----  
Estadísticas  
-----
```

Acciones disponibles:

1. Consultar jugadores por videojuego
2. Consultar estadísticas de las recompensas por videojuego
3. Consultar jugadores que han completado cada videojuego

S. Guardar cambios

Q. Salir

Introduzca una opción:

3.8.1. Consultar jugadores por videojuego

```
-----  
Jugadores con Videojuego N  
-----
```

1. Jugador1
2. Jugador2
3. Jugador3
- ...

Q. Salir

Introduzca una opción:

3.8.2. Consultar estadísticas de las recompensas por videojuego

```
-----  
Estadísticas de las recompensas del videojuego N  
-----
```

1. Recompensa1 20%

2. Recompensa2 80%
3. Recompensa3 5%
- ...

Q. Salir

Introduzca una opción:

3.8.3. Consultar jugadores que han completado cada videojuego

Jugadores que han completado el videojuego N

1. Jugador1
2. Jugador2
3. Jugador3
- ...

Q. Salir

Introduzca una opción:

4. Pruebas del sistema

OP01: createJugador

1. T01: Si se añade un nuevo jugador, se almacena correctamente en el sistema.
2. T02: Ningún campo del jugador se puede almacenar con un valor vacío.
3. T03: No pueden existir dos usuarios con el mismo nickname.

OP02: createVideojuego

1. T04: Si se añade un nuevo videojuego, se almacena correctamente en el sistema.
2. T05: Ningún campo del videojuego se puede almacenar con un valor vacío.
3. T06: Un videojuego debe contener al menos una recompensa.

OP03: deleteVideojuego

1. T07: Si se borra un videojuego se debe borrar correctamente del sistema. También se deben borrar todas las recompensas y DLC asociados a dicho videojuego.

OP04: createRecompensa

1. T08: Si se añade una nueva recompensa, se almacena correctamente en el sistema.
2. T09: Ningún campo de la recompensa se puede almacenar con un valor vacío.

3. T10: Una recompensa debe tener un videojuego asociado.

OP05: deleteRecompensa

1. T11: Si se borra una recompensa se debe borrar correctamente del sistema.

OP06: createDLC

1. T12: Si se borra un DLC se debe borrar correctamente del sistema.

OP07: addJugador

1. T13: Si se añade un nuevo jugador a un videojuego se debe actualizar la lista de jugadores de este videojuego.

OP08: addRecompensa

1. T14: Si un jugador consigue una determinada recompensa se debe actualizar la lista de jugadores que tienen esta recompensa.

5. Entregables

Para evitar que errores en fases tempranas lastren la buena marcha del proyecto, y para mejorar el aprendizaje, no se realizará una única entrega, sino que se realizará una serie de entregas parciales (basándonos en el uso de metodologías de desarrollo ágiles). Para la evaluación continua de la asignatura se deberán cumplir todas las entregas con una calificación suficiente.

La calificación de la práctica se realizará en base a una hoja de cálculo, rúbrica que se hará pública con anterioridad al primer hito. Esta hoja de cálculo se actualizará con la calificación correspondiente a cada versión entregada indicando las posibles mejoras para las siguientes.

A continuación se describen el conjunto de entregas a realizar.

5.1. Primera entrega: Diseño preliminar

- Especificación: Plantear para esta entrega el diseño completo del sistema, obviando los siguientes detalles:
 1. No contemplar la persistencia.
 2. No contemplar las estadísticas (C15, C16 y C17).
- Objetivo: ofrecer una propuesta de diseño sin plantearse la persistencia.
- Formato: Documento de diseño completo.
- Fecha de entrega: **20/11/2018**.

5.2. Segunda entrega: Diseño con Principios

- Especificación: Las mismas que se describieron en el entregable anterior. En este caso, se actualizará el diseño realizado aplicando los principios de diseño OO vistos en la asignatura.
- Objetivo: El mismo que el anterior, especificando de forma justificada los cambios entre el diseño actual y el anterior.
- Formato: Documento de diseño completo.
- Fecha de entrega: **27/11/2018**.

5.3. Tercera entrega: Implementación con memoria

- Especificación: Se implementará el diseño, incluyendo el programa principal, almacenando los datos en memoria. Se valorará que el cambio para la próxima entrega sea mínimo. Se indicarán los patrones de diseño utilizados.
- Objetivo: Además, del diseño detallado se debe entregar el código fuente realizado publicándolo en un repositorio en GitHub.
- Fecha de entrega: **04/12/2018**.

5.4. Cuarta entrega: Implementación persistente

- Especificación: Se añade a la implementación anterior el uso de persistencia, usando una base de datos MySQL.
- Objetivo: Se debe entregar el código fuente actualizado publicándolo en un repositorio Git. Se valorará la automatización en la creación de la base de datos y los campos necesarios. Se debe entregar un manual que indique los comandos a realizar para la creación de la base de datos en otro ordenador diferente.
- Fecha de entrega: **11/12/2018**.

5.5. Quinta entrega: Pruebas unitarias

- Especificación: Añadir las pruebas unitarias en la implementación y pasarlas de forma automática utilizando Travis CI.
- Objetivo: Se debe entregar el código fuente actualizado publicándolo en un repositorio de GitHub. Además se debe indicar el estado correcto en este repositorio vinculándolo correctamente con Travis CI.
- Fecha de entrega: **18/12/2018**.

5.6. Sexta entrega: Final

- Especificación: Completar la implementación anterior añadiendo las estadísticas que se excluyeron desde la primera entrega. Actualizar las pruebas unitarias cubriendo la nueva implementación.
- Objetivo: Al igual que en la entrega anterior, se entregará el código fuente actualizado en un repositorio de GitHub y, además, se indicará el estado correcto de este repositorio haciendo uso de la vinculación existente con Travis CI.
- Fecha de entrega: **15/01/2019**.

6. Evaluación

La evaluación final y defensa de prácticas se realizará el 15 de Enero de 2019. La puntuación se hará de acuerdo a la Tabla [1](#).

<i>Peso</i>	<i>Criterio de evaluación</i>	
15 %	Desarrollo del proyecto	
5 %	Documentación del proyecto	La documentación del proyecto es clara, bien escrita y sin errores. Incluye una introducción y el histórico de cambios.
5 %	Planificación y reparto de tareas	Se identifican las responsabilidades de los miembros del grupo realizadas durante todo el proyecto, debiendo ser equilibrada la asignación de las tareas. Se presenta el cronograma o las diferentes etapas de ejecución del proyecto
5 %	Sistema de control de versiones	Se han usado correctamente a lo largo de todo el proyecto las funciones de svn para la gestión de versiones y mejorar el trabajo en grupo
20 %	Diseño del proyecto	
5 %	Diagrama de clases del proyecto	Se desarrolla correctamente el diagrama de clases de la API, reflejando también las dependencias entre las clases
15 %	Decisiones de diseño	Se describen y justifican las principales decisiones de diseño tomadas. Se identifican y catalogan los patrones de diseño empleados para la mejora del diseño.
20 %	Principios de diseño	
7 %	Ocultación: OCP, ISP, LSP	Los métodos públicos y privados de las clases están bien definidos. Se permite la extensión y el correcto funcionamiento de clases heredadas.
7 %	Cohesión: SRP	Clases bien definidas (SRP) Las clases deben ofrecer funcionalidad concreta y sus métodos mantienen coherencia. Futuras modificaciones no deberían afectar al resto (ortogonalidad).
6 %	DRY	Principio DRY: No debe hacer redundancia en el código, ni de datos ni de cómputo
(+5 % opc.)	POA y Contratos	Se ha utilizado Programación Orientada a Aspectos (con AspectJ) y Contratos (con jContract) (OPCIONAL)
20 %	Calidad del código	
5 %	Control de errores	Correcto control de errores (gestión de excepciones, comprobación de parámetros o empleo de algún mecanismo de log)
5 %	Compleitud	Todos los métodos definidos en la interfaz del API están implementados
5 %	Pruebas unitarias	Se han utilizado pruebas unitarias con Junit durante el desarrollo de la práctica
5 %	Conexión con Sistema de Integración Continua	Se ha conectado correctamente el repositorio con Travis CI y se pasan las pruebas de forma automática, el estado del repositorio es visible y correcto
25 %	Patrones de diseño	
5 %	Patrón creacional	Se utiliza y se justifica al menos un patrón creacional.
10 %	Patrón estructural	Se utiliza y se justifica al menos un patrón estructural.
10 %	Patrón de comportamiento	Se utilizan y se justifican al menos dos patrones de comportamiento.

Cuadro 1: Tabla de evaluación.