



UNIVERSITAT
ROVIRA I VIRGILI

Departament d'Enginyeria Informàtica i Matemàtiques

Proyectos de Sistemas Informáticos

Trabajo en Equipo

Alumno: Dickinson Bedoya Perez

Equipo: 14

Integrantes equipo: Jorge Calzas Moreno,

Erik Giménez Postoyalov,

Nil Pinyol Mateu,

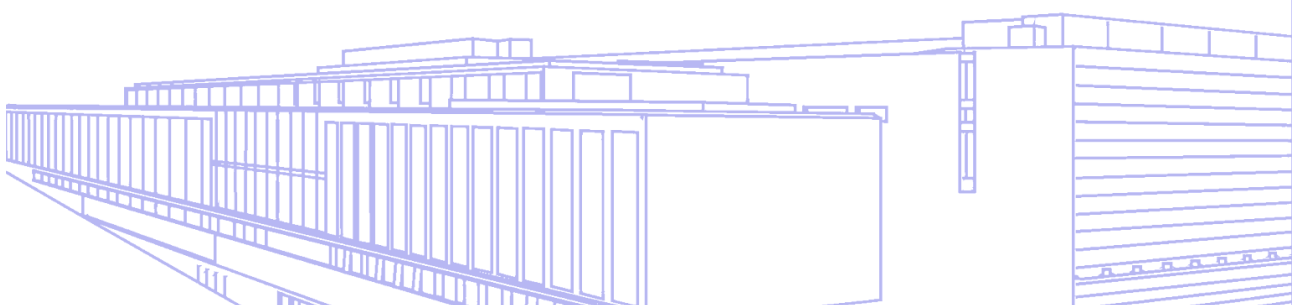
Guillem Rodríguez Mitjana

Asignatura: Proyectos de Sistemas Informáticos

Enseñamiento: Ingeniería Informática

Profesor/es: Susana Prado García, Carlos García Barroso

Fecha: 12/2022



Índice

Introducción	2
Especificaciones del proyecto.....	3
Planificación del proyecto	4
Seguimiento	6
Recepción y evaluación del equipo.....	8
Gestión de la calidad del proyecto desarrollado	10
Evaluación.....	11
Conclusiones	12

Introducción

Presentación del argumento

Este proyecto se basa en la necesidad de poder aprovechar las cosas que ya no se usan en nuestros hogares. Ya hay ciertas plataformas que permiten vender productos de segunda mano, pero lo que buscamos es explorar una visión diferente del mercado. No se quiere vender productos, sino intercambiar estos. Además, se quiere dar una visión amplia del producto ya que no hace falta que sea un objeto físico, sino que también se pueden añadir ciertos servicios.

Motivación

Se ha presentado este proyecto a los alumnos de 2º para que a partir de los conocimientos obtenidos en clase puedan ser capaces de poder desarrollar este proyecto de manera que puedan alcanzar unos conocimientos más avanzados de la programación.

Objetivos del trabajo

Como alumno de PSI, el objetivo claro es poder aprender como poder organizar y guiar a un equipo de trabajo para lograr el desarrollo correcto del proyecto.

Por la parte de los alumnos de la asignatura de programación, el objetivo es poder conseguir que consigan un aprendizaje extenso de este proyecto para poder acabar con conocimientos extensos de la programación orientada a objetos y como se programa con ella.

También, el objetivo entre todos es poder organizarnos como equipo y poder llevar este proyecto a futuro.

Metodología de investigación

Para este proyecto se busca hacer un completo aprendizaje de la programación orientada a objetos y lo que ello conlleva:

- Aprendizaje de tipos de clase:
 - Clase normal, Clase abstracta, Interfaz.
- Aprendizaje de la herencia de clases.
- Aprendizaje de como se generan las listas en Java.
- Aprendizaje de la programación de interfaz gráfica para un usuario.

Especificaciones del proyecto

El proyecto se va a distinguir en la generación de distintas clases:

- **Clase abstracta producto:** Esta clase se va a usar para poder generar las dos clases que heredan de ella. Aquí se definen los atributos principales que van a usar las dos clases herederas para reducir repetición de código.
 - **Clase producto bienes**, hereda de producto: Se definen los atributos y procedimientos específicos de esta clase.
 - **Clase producto servicios**, hereda de producto: Se definen los atributos y procedimientos específicos de esta clase.
- **Clase Lista de productos:** Esta clase pretende generar una lista donde se vayan a guardar todos los productos, es decir, ya sean bienes o servicios se van a guardar en la misma lista. Para ello se hace una clase general para productos y no especifica si es bien o servicio. Con esto se consigue que todos los productos, se guarden en

la misma lista independientemente de que tipo de producto sea. Se tienen que generar los procedimientos específicos de esta clase.

- **Clase Usuario:** Clase que va a tener toda la información relacionada con los usuarios que se vayan a registrar en el sistema. Se tienen que generar los procedimientos específicos de esta clase.
- **Clase Lista de Usuarios:** Clase que pretende generar una lista donde se vayan a almacenar todos los usuarios que estén registrados en el sistema. Esta lista se va a inicializar cuando se inicie el programa según la lectura de un archivo que previamente se habían guardado usuarios.
- **Clase Petición:** Clase específica para tratar las peticiones entre productos. Se debe generar una instancia de esta clase por cada petición que haya, ya sea exitosa o no.
- **Clase Lista Peticiones:** Clase para ir almacenando las peticiones que se vayan generando en la ejecución del programa.
- **Excepciones:** Clases para tratar posibles fallos que surgen en la ejecución del programa, así como lectura de archivos o que las listas estén vacías o llenas.
- **Main:** Se debe generar una clase principal que ejecute el programa según lo esperado, es decir, un programa para el administrador por consola y otro por interfaz gráfica para los usuarios.
- **Otras funcionalidades:** Hay que generar otras funcionalidades para ciertos “*problemas*” que surjan para la ejecución. Una de ellas sería la lectura y escritura de archivos, hay que hacer que esta sea consistente y fiable para no tener fallos a medida que se ejecute el programa.

Planificación del proyecto

Planificación inicial

Planificación hecha en MS-Project, adjuntada en la entrega.

Planificación con actores

Se han definido distintos actores y se ha repartido ciertas tareas para que las haga según su “rol”. Los actores son **programador 1, programador 2, programador 3 y programador 4**. Se les llamará con la abreviatura de “**prog**”.

Clases específicas del **prog1**:

- Clase abstracta producto
- Producto bienes

Clases específicas del **prog2**:

- Clase abst producto
- Producto servicio

Clases específicas del **prog3**:

- Clase usuario
- Lista de usuarios

Clases específicas del **prog4**:

- Clase petición
- Lista de peticiones

Archivos y clases comunes de los programadores:

- Archivos de ejemplo de entrada
- Lectura y escritura de los archivos
 - Escritura de archivos serializados.
 - Escritura de archivos comunes.
- Main
 - Programa principal que controla los menús y que llama a las rutinas.
- Otros
 - Archivos que surjan de soporte para las rutinas implementadas.

Seguimiento

Primera reunión

Se llevan a cabo las estrategias a seguir para afrontar el proyecto.

Todas las tareas se realizan en la fecha y en el tiempo especificado.

Acta de la primera reunión

Fecha: 25/11/2022 9:00 – 10:15

Lugar: Crai Campus Sescelades, sala de trabajo en equipo.

Convocados: Guillem Rodríguez, Erik Giménez, Nil Pinyol, Jorge Calzas, Dickinson Beldoya.

Orden del día:

1. Explicación del proyecto
 - a. Estructura de clases
 - b. Definición de clases y sus métodos
 - c. Metodologías de desarrollo
 - d. Reparto de trabajo
 - e. Sistema de comunicación
2. Asignación de programadores.
3. Confirmación de sistema de comunicación. WhatsApp y Discord.

Primera entrega

Se realiza una reunión antes de la entrega opcional para ver en que parte del proyecto se encuentran.

Se realizan todas las clases básicas para poder tener un feedback de los profesores.

Todas las tareas especificadas para este hito se llevan a cabo en la fecha y tiempo especificado.

Acta de la segunda reunión

Fecha: 02/12/2022 12:30 – 14:30

Lugar: Llamada de discord.

Convocados: Guillem Rodríguez, Erik Giménez, Nil Pinyol, Jorge Calzas, Dickinson Bedyoya.

Orden del día:

1. Seguimiento del proyecto
 - a. Clases básicas ok.
 - b. Lectura y escritura empezado, pero bien encaminado.
 - c. Faltan las listas y tratar ciertos errores.

Entrega final

Se realiza una tercera y última reunión para ver en que punto del proyecto se encuentran y para saber si han alcanzado el objetivo en el tiempo especificado.

Se han implementado ciertos archivos especificados en las tareas de este hito, pero no se llega a realizar ninguna parte de la interfaz gráfica, dejando sin acabar el proyecto en las fechas especificadas.

Se hace énfasis en arreglar ciertas implementaciones que daban conflicto, pero sin dejar de avanzar lo máximo para llegar a la entrega.

Se queda el proyecto con todas las funcionalidades básicas para poder hacer un uso básico en consola, pero como no se implementa la interfaz gráfica el proyecto queda como incompleto en la fecha especificada.

Acta de la ultima reunión

Fecha: 05/01/2023 09:30 – 12:30

Lugar: Llamada de discord.

Convocados: Guillem Rodríguez, Erik Giménez, Nil Pinyol, Jorge Calzas, Dickinson Bedyoya.

Orden del día:

1. Seguimiento del proyecto
 - a. Lectura y escritura ok. Archivos serializados comprobados.
 - b. Listas de objetos probadas y funcionando.

c. Programa principal completo y sólido.

Todo el seguimiento se puede ver el en la planificación hecha en el MS-Project adjunto a este archivo.

Recepción y evaluación del equipo

Entrevista previa

Se ha preparado una reunion con los alumnos de programación para poder explicarles un poco como se va a estructurar el proyecto y como se va a hacer la repartición del trabajo según convenga.

Se ha preparado una pequeña presentación para poder tener gráficamente las ideas a explicar.

Por último, se ha acordado el reparto de trabajo. Para ello se han asignado 4 roles según otras asignaturas para que estén familiarizados con ellos:

- Prog1: Guillem Rodríguez
- Prog2: Erik Giménez
- Prog3: Nil Pinyol
- Prog4: Jorge Calzas

Cada uno de los roles tiene un trabajo asignado explicado en esta reunión y que se ha añadido en la planificación.

Evaluación de las posibles incidencias

Pueden surgir varias incidencias, pero como no son posibles prever todas aquí se van a explicar algunas que pueden ser muy probable que surjan:

1. **Retrasos en la entrega:** Puede haber retrasos en la entrega inicial que pueden venir derivados de diferentes causas:
 - a. **Falta de conocimientos:** Puede ser que los alumnos de programación no hayan entendido bien las explicaciones dadas en clase y no sepan bien como plasmar estas en el proyecto.

- b. **Falta de tiempo:** Es posible que los alumnos no tengan tiempo suficiente para hacer la primera entrega por culpa de tener que prestar atención a otras asignaturas y tener que dejar el proyecto un poco atascado.
- c. **Falta de comunicación:** Puede pasar que haya varias implementaciones hechas y funcionales pero que por falta de comunicación no se indique a otros compañeros que las necesiten y que esto haga un retraso en otras implementaciones.

A medida que se vaya realizando el proyecto puede que surjan otras incidencias que aquí no se han indicado pero que se van a ir solucionando a medida que se pongan delante.

Procedimiento de desarrollo

Para el proyecto se quiere seguir una buena práctica de desarrollo en la programación. Se tienen que seguir los distintos estándares de calidad:

1. Definición de clases y métodos adecuados:
 - a. Seguir convención de nombres:
 - i. Clases en singular.
 - ii. Tipos de nombres:
 1. camelCase
 2. snake_case
 3. PascalCase
 - b. Seguir convenio de Java.
 - i. getUser(), setUser()
2. Código limpio.
 - a. No debe haber exceso de comentarios ni falta de ellos.
 - b. Uso correcto de los espacios y tabulaciones.
3. Gestión correcta de archivos y listas:
 - a. Hace falta que la lectura/escritura de archivos y el tratado de listas sea consistente para evitar fallos en la ejecución del programa.
4. Interfaz gráfica consistente:

- a. La interfaz no debe tener errores así como botones que se mueven o que no son responsivos.
- 5. Juegos de pruebas:
 - a. Los juegos de prueba deben ser rebuscados y extensos para así encontrar todo tipo de problema que haya y poder solucionarlo.
- 6. Documentación extensa:
 - a. Hay que proporcionar una buena documentación del proyecto por parte de los alumnos de programación. A parte de redactarla, deben usar la herramienta **JavaDoc** para que se generen las documentaciones pertinentes de las clases.

Herramientas de trabajo

Se van a usar distintas herramientas de trabajo para poder colaborar entre el grupo:

1. **Comunicación:** Para comunicarnos usaremos distintas plataformas.
 - a. Tablero del Moodle
 - b. WhatsApp
 - c. Discord
 - d. Reunion presencial
2. **Documentación:** Para poder documentar el proyecto se va a usar las siguientes opciones:
 - a. Word
 - b. Google Docs. Mejor opción ya que es muy colaborativo.
 - c. JavaDocs
3. **Herramientas de desarrollo:** Para poder desarrollar el proyecto se va a usar los siguientes programas y sus versiones:
 - a. **Java** versión 19.0.1
 - b. Eclipse versión 4.25
 - c. Github. En el proyecto no se va a usar pero es una buena opción al desarrollo.

Gestión de la calidad del proyecto desarrollado

Para probar la calidad del desarrollo del proyecto se han aplicado los siguientes aspectos:

- Pruebas unitarias para hacer testing de todas las clases desarrolladas:

- Lectura correcta de archivos y generado de instancias para poder ejecutar el programa. Se tiene que ver si:
 - Se cargan correctamente los usuarios leídos del archivo.
 - Se cargan bien los productos.
 - Se cargan bien las peticiones.
 - Se asignan bien las peticiones y los productos de los usuarios.
- Operar con las clases:
 - Se instancian bien los objetos de las clases básicas.
 - Se instancian bien las clases de las listas.
 - Se añaden bien las instancias de objetos a la lista de la clase específica.
 - Se buscan, editan y borran objetos dentro de las listas de objetos.

Las pruebas unitarias serán realizadas según el programador que ha hecho la clase. En cualquier caso, los otros programadores deberán ayudar a ver aspectos que el propio desarrollador del test no ha sabido ver.

Evaluación

Descripción	OK?
Clase Producto	Funciona correctamente
Clase Producto Bienes	Hereda y funciona correctamente
Clase Producto Servicios	Hereda y funciona correctamente
Clase Usuarios	Funciona correctamente
Clase petición	Funciona correctamente
Clase Lista de productos	Funciona correctamente
Clase Lista de usuarios	Funciona correctamente

Clase Lista de peticiones	Funciona correctamente
Lectura correcta de archivos	Funciona correctamente
Cargado correcto de datos después de leer de un archivo	Funciona correctamente
Obtener datos específicos de una lista de objetos	Funciona correctamente
Borrar datos específicos de una lista de objetos	Funciona correctamente
Editar datos específicos de una lista de objetos	Funciona correctamente
La interfaz gráfica es solida	No implementado
Los botones de la interfaz gráfica no dan fallos	No implementado
Los datos que muestra la interfaz son correctos	No implementado

Conclusiones

La evidencia que presento anteriormente demuestra que el proyecto se ha realizado según los criterios básicos demandados por la asignatura. Aunque no se haya realizado la parte gráfica el funcionamiento básico del programa es correcto.

Como director del equipo de unos alumnos con nociones básicas de programación en Java, debo decir que el simple hecho de ir comentando los aspectos en los que se iban quedando enganchados ayudaba a las dos partes para seguir continuando, es decir, yo sé en que punto del proyecto van y ellos saben como resolver el problema, haciendo que sea productivo para todo el equipo.

Comentando aspectos del equipo de programación, debo decir que me ha faltado mas respuestas de ciertos integrantes del equipo, solo han hablado dos de ellos y los que no lo han hecho en momentos puntuales cuando les preguntaba aspectos que los implicaba, como su estado en el proyecto. A parte, hay falta de comunicación entre ellos por lo que ha dificultado ciertos aspectos en el desarrollo.

También quiero comentar que han pasado de no tener casi nociones por no decir ninguna de la programación en Java a poder desenvolverse en el código.

En consideración de las asignaturas de PSI y Programación se pueden destacar ciertos aspectos en los que se está conforme y desconforme.

En la asignatura de PSI, se debe decir que el hecho de gestionar un proyecto con unos alumnos de Programación hace que la asignatura sea más entretenida y enriquecedora. Dejando de lado otros aspectos de la asignatura, lo mejor ha sido poder ser el líder que un alumno de PSI lo fue con mis compañeros hace un par de años y poder guiar y conocer a otros alumnos en su transcurso en la asignatura.

En lo que consta a la asignatura de Programación, se debe decir que se podría usar otras IDE más actuales y actualizadas para que sean conscientes de las nuevas tendencias en el mundo de la informática.

En el equipo hemos decidido hacer un repositorio de GitHub en el que se van a ir teniendo un control de las versiones. Esto se podría haber exigido por parte de la asignatura de programación ya que genera facilidad en ver como se va avanzando en el proyecto. Además de hacer que se acostumbren a usarlo ya que es una herramienta que llega a ser usada por la mayoría de los proyectos informáticos.

En cuanto a los conceptos aprendidos, los alumnos ya saben los conceptos básicos de la programación orientada a objetos y pueden desenvolverse fácilmente en los nuevos aspectos relacionados con la programación.

En definitiva, las asignaturas hacen ver como es gestionar y hacer un proyecto de manera más agradable y práctica, pero tiene aspectos en los que se podrían mejorar las dos.

- Se podría organizar mas antes de empezar la asignatura
- Se deberían usar más tecnologías nuevas que ayuden al desarrollo, así como repositorios de la universidad para tener un control de ellos y nuevas interfaces.