

PRÁCTICA 2

Roles Scrum y Plan de Gestión de Configuración Software



The Video Game Box

INTEGRANTES DEL GRUPO:

MARIO CAMPOS SOBRINO
CARLOS CARNERO MÉRIDA
JOSÉ DÍAZ REVIEJO
DAVID ELÍAS PIÑEIRO
CARLOS GÓMEZ LÓPEZ
ÁLVARO GÓMEZ SITTIMA
JULIÁN MOFFATT
JUAN ROMO IRIBARREN
JAVIER DE VICENTE VÁZQUEZ
GONZALO VÍLCHEZ RODRÍGUEZ

0. Índice

1.	Roles Scrum	3
1.1	Mario Campos Sobrino.....	3
1.2	Carlos Carnero Mérida	3
1.3	José Díaz Reviejo	3
1.4	David Elías Piñeiro	3
1.5	Carlos Gómez López.....	3
1.6	Álvaro Gómez Sittima.....	4
1.7	Julián Moffatt	4
1.8	Juan Romo Iribarren	4
1.9	Javier De Vicente Vázquez.....	4
1.10	Gonzalo Vilchez Rodríguez	4
2.	Plan de Gestión de Configuración	5
2.1	Productos y Procedimientos	5
2.1.1	Acta de Reuniones.....	5
2.1.2	Plan de Proyecto Software.....	6
2.1.3	Código Fuente y Ejecutable.....	6
2.1.4	Manual de Usuarios.....	6
2.1.5	Burndown Chart	6
2.1.6	Story Map	6
2.1.7	Daily Scrum	6
2.1.8	Product Backlog	6
2.1.9	Sprint Backlog	7
2.1.10	Sprint Planning	7
2.1.11	Sprint Review	7
2.1.12	Sprint Retrospective	7
2.1.13	Impediment Backlog	7
2.2	Herramientas y ramas.....	7
2.2.1	Jira	8
2.2.2	GitHub.....	8
2.2.3	Eclipse/IntelliJ.....	8
2.2.4	IBM	8
2.2.5	MongoDB	8
2.2.6	AWS	8
2.2.7	API de Steam.....	8
2.2.8	API de GOG.....	8
2.2.9	API de IGDB.....	9
2.2.10	Microsoft Office.....	9
2.2.11	OneDrive	9
2.2.12	Hibernate	9
2.2.13	Spring	9
2.3	Estructura de carpetas	9
2.4	Nomenclatura.....	10
2.5	Permisos en el trabajo	11
3.	Parte Opcional	12
3.1	Team Building	12

1. Roles Scrum

El primer aspecto que vamos a tratar es el reparto de los roles Scrum. Para llevar a cabo este reparto nos hemos basado en los roles de equipo asignados en el proyecto y también en las habilidades que pueden aportar.

A continuación, se justificarán los roles Scrum asignados a cada integrante y se mostrarán en la [tabla 1.11](#).

1.1 Mario Campos Sobrino

Mario formará parte del Equipo de Desarrollo, por 2 razones. Una de ellas es que personalmente siente que realizará mejor trabajo perteneciendo al equipo de desarrollo, y la segunda y más principal es por descarte, ya que en los otros roles Scrum encajan mejor otros compañeros.

1.2 Carlos Carnero Mérida

A Carlos se le ha asignado formar parte del [Equipo de Desarrollo](#). El motivo fundamental de esta decisión es su rol de equipo, *Finalizador*, el cual nos asegura que siempre se entregará de manera correcta y en tiempo todo lo que el proyecto va requiriendo y cuidando siempre que no se pasen por alto detalles importantes. Además, destaca su facilidad para trabajar en equipo y su flexibilidad para llevar a cabo las diferentes tareas necesarias.

1.3 José Díaz Reviejo

José va a ejercer el rol de [Scrum Master](#) puesto que dentro del equipo tiene el rol de Coordinador, además está dispuesto a entender Scrum y hacérselo llegar al resto del equipo, puesto que es nuevo para todos.

Hemos elegido a este miembro como Scrum Master debido a sus habilidades sociales y su actitud en las primeras reuniones, donde se mostraba participativo y capaz de guiar una reunión de varias personas.

1.4 David Elías Piñeiro

David va a formar parte del [Equipo de Desarrollo](#), pues los roles que obtuvo como resultado del test de Belbin son principalmente de acción (Implementador e Impulsor). Gracias a su rol implementador, desarrollará e implementará lo que se necesite hacer independientemente de que le guste o no la tarea. Al ser también impulsor nos aseguramos tener un miembro que trabaje bien bajo presión y sea capaz de sacar adelante tareas que han podido retrasarse.

1.5 Carlos Gómez López

A Carlos se le ha asignado al [Equipo de Desarrollo](#). La razón más importante es por el rol de equipo que desempeña, que es *Finalizador*, ya que su rol nos va a asegurar que se entregará una versión del proyecto 100% funcional. Además, nos ha confirmado que sus puntos fuertes son el diseño gráfico y la codificación.

1.6 Álvaro Gómez Sittima

Para Álvaro se le ha asignado el rol de *Equipo de Desarrollo*, principalmente por preferencias, y también por descarte ya que no cree que vaya a hacer tan buen trabajo de Product Owner o de Scrum Master como otros compañeros, pero también debido a que como sus roles asignados son Implementador y Especialista puede asegurar que las ideas propuestas se implementen correctamente, ayudando a otros miembros del equipo de desarrollo si fuese necesario.

1.7 Julián Moffatt

Julián se ofreció voluntario desde el primer momento para desempeñar el rol de *Product Owner*, además tanto su rol dentro del equipo (Cohesionador) como su personalidad (Protagonista) son adecuados para mantener una buena comunicación con el equipo de desarrollo cada vez que reciba una entrega al finalizar cada sprint.

Julián ha entendido el producto y ha aportado nuevas ideas por lo que creemos que es el indicado para ejercer de Product Owner.

1.8 Juan Romo Iribarren

Juan va a formar parte del *Equipo de Desarrollo*. Las ventajas que puede aportar se deben a su rol de Cerebro, ya que podrá ayudar al resto del Equipo de Desarrollo aportando propuestas e ideas creativas que hagan del producto algo innovador, pero dentro de lo solicitado. También es capaz de analizar de una manera crítica y lógica si se está realizando un trabajo de la manera más efectiva posible o dar una vuelta de tuerca a ideas que se pueden mejorar.

1.9 Javier De Vicente Vázquez

Javier formará parte del *Equipo de Desarrollo*, en parte por preferencias personales ya que se considera que será más útil al equipo desarrollando la aplicación y en parte por descarte ya que Julián y José son los candidatos idóneos para ser Product Owner y Scrum Master respectivamente.

1.10 Gonzalo Vílchez Rodríguez

En el caso de Gonzalo, también participará dentro del proyecto con el rol de *Equipo de Desarrollo* ya que sus roles principalmente son de acción. Destaca su rol de Especialista que puede servir para ayudar a los demás integrantes con algunos problemas del desarrollo, a la vez que tiene el rol de implementador para ayudar a codificar. Con su rol de *Monitor Evaluador* también puede ayudar al equipo diagnosticando algunos problemas para poder encontrar soluciones.

Integrantes	Roles Scrum
Mario Campos Sobrino	Equipo de Desarrollo
Carlos Carnero Mérida	Equipo de Desarrollo
José Díaz Reviejo	Scrum Master
David Elías Piñeiro	Equipo de Desarrollo
Carlos Gómez López	Equipo de Desarrollo
Álvaro Gómez Sittima	Equipo de Desarrollo
Julián Moffatt	Product Owner
Juan Romo Iribarren	Equipo de Desarrollo
Javier De Vicente Vázquez	Equipo de Desarrollo
Gonzalo Vílchez Rodríguez	Equipo de Desarrollo

Tabla 1.11 Roles Scrum

2. Plan de Gestión de Configuración

2.1 Productos y Procedimientos

Una vez repartido los roles Scrum entre los integrantes, es el momento de desarrollar un plan de gestión de configuración en el proyecto.

La *gestión de la configuración* es el conjunto de procesos destinados a asegurar la calidad de todo producto obtenido durante cualquiera de las etapas del desarrollo de un sistema de información, a través del estricto control de los cambios realizados sobre los mismos y la disponibilidad constante de una versión estable de cada elemento para toda persona involucrada en el desarrollo. Toda esta información es creada como parte del proceso de Ingeniería de Software.

A continuación, se definen tanto los productos que se pondrán bajo control de configuración como los procedimientos que deben ser seguidos por los integrantes del equipo, es decir, los elementos de la configuración del software (ECSs):

2.1.1 Acta de Reuniones

Consiste en un documento que registra los temas tratados y los acuerdos adoptados en las reuniones, con la finalidad de certificar lo acontecido y dar validez a lo acordado. En nuestro caso solo registraremos las conclusiones acordadas en la reunión para simplificar y porque al seguir la metodología agile no es viable registrar todos y cada uno de los detalles acordados dado al volumen de reuniones que se desarrollan. Se dejará escrito tanto la fecha, como el número del Sprint en el que nos encontramos, así como los asistentes a la reunión, además de la duración de la reunión junto con las conclusiones a las que se han llegado. Todo ello quedará guardado en un archivo Word en la carpeta designada en nuestra estructura de carpetas.

2.1.2 Plan de Proyecto Software

El plan de proyecto software se encarga de englobar la información relativa a la gestión del proyecto. Éste no será excesivamente extenso ni complicado, pues debe servir como guía al equipo de desarrollo, respondiendo a las cuestiones referentes a cada aspecto del proyecto de forma específica. Plasma todos los recursos disponibles para realizar el desarrollo, y los asigna a tareas concretas, con fechas y mecanismos concretos de control y seguimiento. El plan de proyecto lo desarrolla el *Scrum Master*, guardará la información en un archivo Word en la carpeta designada en nuestra estructura de carpetas.

2.1.3 Código Fuente y Ejecutable

El código fuente se realizará por parte del Equipo de Desarrollo, y el Ejecutable se ejecutará uno por cada sprint y permitirá iniciar el programa.

2.1.4 Manual de Usuarios

Es un documento técnico que explica el funcionamiento del programa. Aquí se exponen los procesos que el usuario puede realizar con el sistema implementado. Para ello se detallarán todas y cada una de las características que tiene el programa. Debe ser el documento definitivo de cara al usuario.

2.1.5 Burndown Chart

Es una forma de medida gráfica del progreso de un proyecto Scrum. El Scrum Master será el encargado de actualizar este gráfico al finalizar cada sprint. En el eje horizontal se muestra el número de sprints y en el eje vertical muestra la cantidad de trabajo pendiente por realizar al inicio de cada sprint, que se medirá en Story Points.

2.1.6 Story Map

Es una técnica colaborativa de ayuda visual que utilizan tanto el Product Owner como el equipo de desarrollo, actuando el Scrum Master como un moderador. Mostrará los flujos de trabajo reales, ofreciendo una vista general de todas las funcionalidades del producto de punta a punta. Además, será útil para tomar decisiones sobre el producto.

2.1.7 Daily Scrum

Es una reunión diaria de unos 15 minutos durante el desarrollo de un sprint en la que cada miembro del equipo de desarrollo contará lo que hizo el día anterior, los problemas que se encontró y como los resolvió, además de lo que va a hacer ese día.

2.1.8 Product Backlog

Consistirá en una lista ordenada por prioridad de todo lo que el Product Owner considera que el producto debe tener. Éste se encargará de disponer en la lista las funcionalidades desde las imprescindibles, las que el producto debe tener si o si, hasta los detalles totalmente prescindibles. Al ser un

documento "vivo", se irá actualizando continuamente teniendo en cuenta nuevas necesidades, cosas que no son necesarias y el plazo para realizar el proyecto. Todo lo prescindible no se implementará a menos que se disponga del tiempo suficiente para ello. El Product Owner tendrá que hablar continuamente con el equipo de desarrollo sobre lo que se irá implementando.

2.1.9 Sprint Backlog

El Product Owner y el Equipo de Desarrollo eligen los ítems del Product Backlog que desean desarrollar en el sprint (Sprint Planning). El equipo de desarrollo descompone estos ítems en tareas que formarán el Sprint Backlog. Se distribuirán entre tres listas durante el sprint. Las tareas por realizar (to do), las que se están realizando (doing) y las realizadas (done). Esto permitirá a los integrantes del equipo de desarrollo saber que tareas no se han realizado aún y organizarse durante el sprint.

2.1.10 Sprint Planning

El martes en el laboratorio se reúnen el Product Owner y el Equipo de Desarrollo para decidir las funcionalidades que se van a desarrollar en el próximo sprint. Se comprometen a desarrollar lo que realísticamente podrá ser entregado, funcionando y testeado al finalizar el sprint. Se discuten y eligen las funcionalidades que el Product Owner considera más prioritarias.

2.1.11 Sprint Review

El martes en el laboratorio el Equipo de Desarrollo presenta al Product Owner el producto para revisarlo y obtener un feedback. Es una reunión centrada en el producto, en la que se hablará sobre si el resultado es satisfactorio o algo que no se pudo implementar como fue planeado. Cada sprint durará 2 semanas.

2.1.12 Sprint Retrospective

El martes en el laboratorio se revisa el proceso llevado a cabo en el último sprint y se revisa buscando que en el próximo sprint se trabaje de manera óptima. Los integrantes del proyecto dirán que consideran que se debe cambiar o es mejorable, aportando ideas para tomar decisiones con el propósito de mejorar la dinámica de los sprints.

2.1.13 Impediment Backlog

El equipo listará en un documento las situaciones que impiden que se progrese durante el sprint. El Scrum Master es el responsable de solucionarlas en las 24h posteriores a la identificación. Se realizará uno por cada sprint.

2.2 Herramientas y ramas

Para almacenar los datos, ayudarnos a obtener información, documentar el proyecto y codificarlo, vamos a emplear distintas herramientas que hemos seleccionado para la gestión de configuración. A continuación, vamos a justificar por qué hemos elegido estas herramientas.

2.2.1 Jira

Jira es una herramienta de gestión de proyectos de metodologías ágiles, es el más usado a nivel profesional por lo que nos interesa empezar a familiarizarnos con este tipo de aplicaciones. Otro factor importante es la gran integración que tiene con aplicaciones de terceros lo cual nos abre un gran abanico de opciones. Por último, tiene una licencia gratuita para grupos de no más de 10 personas, adecuándose perfectamente a nuestras necesidades.

2.2.2 GitHub

Es el repositorio más usado a nivel profesional y no profesional con gran diferencia, a parte todos los miembros del equipo lo saben usar en mayor o menor medida.

2.2.3 Eclipse/IntelliJ

Son dos IDEs con las que estamos familiarizados pero cada persona tiene libertad para elegir en entorno que más se adapte a sus preferencias.

2.2.4 IBM

Como su propio nombre dice, es una aplicación de diseño de software, especialmente enfocada a diagramas UML. Por su uso en asignaturas previas a GPS todos los miembros del equipo están muy familiarizados con ella y la hace la mejor opción.

2.2.5 MongoDB

Es una base de datos NoSQL basada en ficheros lo cual nos llama mucho la atención ya que es algo con lo que no estamos familiarizados y nos supondrá un reto, pero nos gusta la idea de aprenderla, ya que hasta el momento solo hemos estudiado bases de datos SQL.

2.2.6 AWS

La base de datos estaría basada en la herramienta ofrecida por MongoDB, que a su vez se hostea en los servidores de AWS que son los más usados a nivel profesional y su fiabilidad esta más que probada lo que los hace mejor opción, aparte de tener un apartado de opciones gratis por lo que la base de datos sería hosteada de forma gratuita.

2.2.7 API de Steam

Steam tiene una Web API basada en HTTP que se puede usar para acceder a muchas características de Steamworks. A través de los métodos que contiene podremos validar las credenciales de Steam para así poder vincular una cuenta con la aplicación y poder obtener información necesaria de los juegos.

2.2.8 API de GOG

GOG tiene una API que permitirá acceder a la información de los usuarios que se encuentran en la plataforma GOG.

2.2.9 API de IGDB

También se usará la API de IGDB de la cual podremos extraer los datos necesarios para poder construir la base de datos.

2.2.10 Microsoft Office

Lo utilizamos para la documentación, ya que Office es la aplicación más usada para este caso y la UCM nos da una licencia gratuita por lo que todos los miembros del equipo tienen acceso a ella al completo.

2.2.11 OneDrive

Utilizamos OneDrive para escribir documentos en tiempo real, editar documentos de forma simultánea ya que en nuestra opinión tiene más opciones y funciona mejor que otras herramientas como Google Drive. Aparte que está basado en Office, la aplicación que usaremos para la documentación.

2.2.12 Hibernate

Es una herramienta de mapeo de objeto-relacional (ORM). Es una de las herramientas más utilizadas para facilitar el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación.

2.2.13 Spring

Es una herramienta que da soporte a Hibernate por lo que se puede utilizar como ORM, pero también ofrece otras utilidades para desarrollo de aplicaciones web.

Estas dos últimas herramientas son muy conocidas y utilizadas, por lo tanto, nos llaman la atención, pero como en un principio solo vamos a desarrollar una aplicación de escritorio y no se asegura el que lleguemos a hacer también una aplicación web no se ha decidido cuál de las dos herramientas (Hibernate o Spring) vamos a acabar utilizando

2.3 Estructura de carpetas

A continuación, nos vamos a centrar en cómo vamos a estructurar las carpetas dentro de nuestro proyecto, para así tener un trabajo organizado y a la hora de trabajar, ser más eficientes. La estructura elegida se visualizará en la [Imagen 2.3.1](#).

Para poder organizarnos vamos a tener una carpeta principal del proyecto que se llama [Proyecto](#), la cual va a contener otras carpetas que vamos a ir definiendo a continuación.

La primera subcarpeta sería [Código](#). Esta carpeta albergará el código fuente dividido en [Presentación](#), que contendrá el código relacionado con las vistas e interfaces de la aplicación, también [Negocio](#), carpeta que albergará el código relacionado con los servicios del sistema, y, por último, [Pruebas](#), carpeta donde se encontrarán las pruebas automatizadas.

La segunda subcarpeta se llama *Diseño*. En esta carpeta se guardarán todos los diagramas del proyecto, dividiéndose en *Modelo del Dominio* (diagramas relacionados con la estructura de base de datos), *Diagramas de Clase* y *Diagramas de Actividad*.

También tendrá otra subcarpeta definida como *Documentación*, en esta carpeta se guardarán todos los documentos para informar del proceso Scrum, estando dividida en subcarpetas por *artefactos Scrum*.

Otra subcarpeta sería *Reuniones*, la cual albergará las grabaciones de vídeo y conclusiones de las reuniones realizadas a lo largo del proyecto (Daily Scrum, Reunión de revisión, Reunión de planificación, Reunión de retrospectiva), a partir del inicio del primer sprint.

Finalmente, la última subcarpeta a definir es *Recursos*. Una carpeta que va a contener los elementos disponibles para el correcto funcionamiento del proyecto (imágenes, logos, instaladores de herramientas que utilicemos, controladores, etc.).

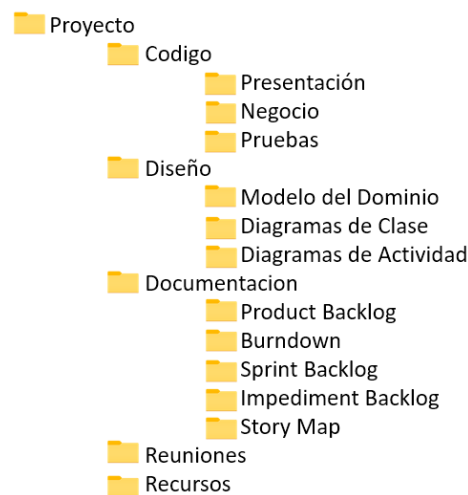


Imagen 2.3.1 Estructura de carpetas del Proyecto

2.4 Nomenclatura

Para mayor comodidad a la hora de trabajar, asignamos a cada elemento de configuración software una nomenclatura, la cual nos permitirá ser más eficientes a la hora de búsqueda de información o de escritura de datos.

Con respecto a la Reuniones, todas las reuniones, sean del tipo que sean, se encontrarán en la misma carpeta por lo tanto éstas se identificarán mediante la fecha y tipo de reunión: Fecha + Tipo de Reunión. También se emplearán abreviaturas como: DS (Daily Scrum), SReview (Sprint Review), SP (Sprint Planning), SRetrospective (Sprint Retrospective).

Para los elementos de la configuración del software, como solo tenemos un proyecto, podemos identificar los ECSs mediante un identificador único para

cada ECSs distinto y su versión. Ejemplos: PlanProyecto_V2.0.1 , ProductBacklog_V1.

Para el resto de las nomenclaturas asignadas, van a ser visiblemente representadas en la [tabla 2.4.1](#).

CARPETA PRINCIPAL	SUBCARPETA	ECS	FORMATO
Proyecto	Documentación	Plan de proyecto	PlanProyecto + <Versión>
Proyecto	Documentación	Plan de Gestión de Configuración	PlanGestionConfiguracion + <Versión>
Proyecto	Documentación	Manual de usuario	ManualUsuario + <Versión>
Proyecto	Documentación/BurnDown	Burndown chart	BurndownChart + <Versión>
Proyecto	Documentación/StoryMap	Story Map	StoryMap + <Versión>
Proyecto	Documentación/ProductBacklog	ProductBacklog	ProductBacklog + <Versión>
Proyecto	Documentación/SprintBacklog	Sprint Backlog	SBacklog + <Versión>
Proyecto	Documentación/ImpedimentBacklog	Impediment Backlog	ImpedimentBacklog + <Versión>
Proyecto	Código	Código fuente	<Módulo> + <Funcionalidad>
Proyecto	Reuniones	Sprint Planning	<Fecha> + SP
Proyecto	Reuniones	Sprint Review	<Fecha> + SReview
Proyecto	Reuniones	Sprint Retrospective	<Fecha> + SRetrospective

Tabla 2.4.1 Nomenclatura ECSs

2.5 Permisos en el trabajo

Tras ver las herramientas que vamos a emplear, tenemos que dar permisos de lectura, escritura y borrado a cada integrante en función del rol que les haya sido asignado por cada elemento de configuración de control, siendo visible en la [tabla 2.5.1](#).

A la hora de asignar los permisos a los integrantes, los vamos a diferenciar en función de sus roles, por ello para el [Scrum Master](#), va a ser el que va a tener acceso a la mayoría de los permisos de los ECSs, ya que en sí se va a encargar sobre todo de la zona de documentación y de realizar, editar y dirigir las reuniones del equipo (Sprint Retrospective, Sprint Planning, Sprint Reviews, Acta de Reuniones y Daily Scrum), pero con respecto al código y diseño, solo va a tener permisos de lectura, ya que quienes van a tener todos los permisos van a ser el [Equipo de Desarrollo](#). El Equipo de Desarrollo también va a encontrar permisos en los recursos, porque van a tener que hacer uso de ellos, y dentro

de documentación en los *StoryMap* van a tener todos los permisos , porque van a ser responsables de ellos, a diferencia del Scrum Master, que no va a tener el permiso de *Destroy*.

Por último, el *Product Owner*, el único ECS en el que va a tener todos los permisos va a ser el *Product Backlog*, porque va a ser el propio Product Owner quien va a controlar dicha lista ordenada de sus prioridades, estando habitualmente en contacto con el Equipo de Desarrollo.

CARPETA PRINCIPAL	SUBCARPETA	SCRUM MASTER				DESARROLLADORES				PRODUCT OWNER			
		Read	ChkIn ChkOut	Add Ren Del	Destroy	Read	ChkIn ChkOut	Add Ren Del	Destroy	Read	ChkIn ChkOut	Add Ren Del	Destroy
Proyecto	Código	✓	X	X	X	✓	✓	✓	✓	✓	X	X	X
Proyecto	Diseño/ModeloDelDominio	✓	X	X	X	✓	✓	✓	✓	✓	X	X	X
Proyecto	Diseño/DiagramasDeClase	✓	X	X	X	✓	✓	✓	✓	✓	X	X	X
Proyecto	Diseño/DiagramasDeActividad	✓	X	X	X	✓	✓	✓	✓	✓	X	X	X
Proyecto	Documentacion/ProductBackLog	✓	✓	✓	✓	✓	X	✓	X	✓	✓	✓	✓
Proyecto	Documentacion/BurnDown	✓	✓	✓	✓	✓	✓	✓	X	✓	X	X	X
Proyecto	Documentacion/SprintBackLog	✓	✓	✓	✓	✓	X	X	X	✓	X	X	X
Proyecto	Documentacion/ImpedimentBacklog	✓	✓	✓	✓	✓	X	X	X	✓	X	X	X
Proyecto	Documentacion/StoryMap	✓	✓	✓	X	✓	✓	✓	✓	✓	X	X	X
Proyecto	Documentación/N/A	✓	✓	✓	X	✓	X	X	X	✓	X	X	X
Proyecto	Reuniones	✓	✓	✓	✓	✓	X	X	X	✓	X	X	X
Proyecto	Recursos	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	X	X

Tabla 2.5.1 Permisos

3. Parte Opcional

3.1 Team Building

Como actividad grupal nos reunimos a comer y tomar algo al terminar las clases en la facultad.

La actividad surgió con el objetivo principal de conocernos todos mejor y en persona ya que hasta el momento hemos trabajado principalmente de manera online y muchos de nosotros no nos conocíamos aún fuera del aula. Además, queríamos también con ello evitar un posible conflicto interno en el grupo debido a ciertas discrepancias con las notas de la primera práctica.

Durante la dinámica charlamos durante varias horas en la terraza de la facultad sobre temas diversos. A pesar de que en el vídeo solo consta media hora por problemas de memoria, estuvimos varias horas en las que tanto abordamos temas importantes como lo son para nosotros la diferencia en las notas, como también lo pasamos bien y nos conocimos bastante más los unos a los otros.

Como análisis final de esta dinámica, creemos que ha sido muy positiva para nosotros ya que nos conocimos más a fondo todos y cumplimos por tanto nuestro objetivo principal, mejorando así la comunicación dentro del equipo y sobre todo la confianza entre nosotros. Además, al resolver también el conflicto de las notas, esto nos ha ayudado a mejorar el ambiente en el grupo y estar más motivados y unidos de cara a próximas entregas.

A continuación, se encuentra el vídeo mencionado de una parte de lo que fue nuestra reunión.

