MEMORIA PRÁCTICA 2 MMIT

CURSO 2017-2018



**INTRODUCCIÓN**

[1. Introducción 1](#_Toc508546030)

[2. Historias de usuario 2](#_Toc508546031)

[a) Herramientas usadas para gestionar el P.B. y el S.B. 3](#_Toc508546032)

[b) Validación de las Historias de Usuario 3](#_Toc508546033)

[C) PRIORIZACIÓN DE LAS HISTORIAS DE USUARIO 4](#_Toc508546034)

[D) MOCKUPS CREADOS 4](#_Toc508546035)

[e) Perfiles de usuario 4](#_Toc508546036)

[3.-Prototipo 5](#_Toc508546037)

[A) HERRAMIENTAS Y TECNOLOGÍAS PROBADAS 5](#_Toc508546038)

[B) DESARROLLO DEL PROTOTIPO 6](#_Toc508546039)

[C) INSTRUCCIONES DE EJECUCIÓN Y CÓMO PROBAR EL PROTOTIPO 8](#_Toc508546040)

[4.- Dinámica grupal 9](#_Toc508546041)

[A) DESCRIPCIÓN DE LA DINÁMICA 9](#_Toc508546042)

[B) DESARROLLO 10](#_Toc508546043)

[C) RESULTADOS 10](#_Toc508546044)

# **Introducción**

Inicialmente, nos pusimos a debatir sobre qué herramientas serían más convenientes para el *Product Backlog* y el *Sprint Backlog*.

Tras haber acordado que usaríamos *Jira* para gestionar el *Product Backlog* y el *Sprint Backlog*, el equipo se dividió en dos grupos. Uno de ellos estaba formado por el ***Product Owner* y el *Scrum Master***, que se dedicaron a escribir sobre las historias de usuario que formarían posteriormente el *Product Backlog*.

Al principio, quedaron bastantes Historias de Usuario épicas, pero con ayuda de todo el grupo se fueron simplificando y dividiendo en unas más simples, que serían sobre las que se harían los criterios de aceptación posteriormente.

Cuando estuvieron redactadas las Historias de Usuario, nos juntamos todo el grupo para validarlas. Esto resultó ser ligeramente arduo, pues teníamos algunos conceptos erróneos sobre cómo realizar las Historias de Usuario, además de que salieron algunas épicas.

Tras ser validadas por todo el equipo, serían acompañadas de los criterios de aceptación correspondientes. Así, conseguimos tener las Historias de Usuario listas para ser priorizadas.

Para la priorización, se decidió usar el método *Kano,* pues era simple de aplicar, y nos parecía más objetivo que otros como el de los *100 puntos*. Cuando el *Product Owner* hubo priorizado todas las Historias de Usuario, solo nos quedaron por realizar las partes opcionales de la práctica.

El otro grupo, que estaba formado por los **desarrolladores**, se dedicó a practicar con las herramientas y tecnologías que se usarían en el proyecto (mientras el Product Owner y el Scrum Master elaboraban las Historias de Usuario) para más tarde crear el prototipo.

Se acordó que el prototipo tendría varias funciones básicas que permitiera tanto manejar la parte de interfaces gráficas de Netbeans, como crear y mantener una base de datos con SQLite (previamente creada) en lenguaje Java. Además, habría que hacer un *login* al iniciar el prototipo.

Tras probar el prototipo, analizar y corregir errores, se preparó para que se pudiera ejecutar desde un *.jar* y generara automáticamente la base de datos.También se elaboraron las instrucciones de ejecución correspondientes, y se redactó las dificultades encontradas a la hora de realizar el prototipo.

Por último, se añadieron las partes opcionales a la práctica. Se realizó una dinámica grupal que será explicada posteriormente, así como mockups y perfiles de usuarios que servirían para tener un boceto de lo que sería la aplicación, y poder meternos en el papel de los distintos usuarios de nuestra aplicación.

# **Historias de usuario**

En los siguientes apartados explicaremos las herramientas usadas para que el Product Owner y el equipo de desarrollo puedan gestionar el Product Backlog y el Sprint Backlog, respectivamente.

También se explicará el proceso de validación de las Historias de Usuario, y cómo el Product Owner las priorizó posteriormente.

Como añadido, hemos incluido los Mockups de la aplicación, y perfiles de usuario.

## **a) Herramientas usadas para gestionar el P.B. y el S.B.**

Para el *Sprint Backlog* y *Product Backlog* decidimos usar **Jira**.

Una de nuestras opciones era usar *Trello* para la gestión del Sprint Backlog, debido a que su uso era sencillo. Sin embargo, Jira tiene más funcionalidades, tiene un uso bastante extendido, y con otras herramientas como GitHub (que también usaremos, pero para el control de versiones), es más difícil dividir el *Product Backlog* en tareas.

Primero probamos *Trello*, que resultaba ser muy amigable para el usuario, pero no tenía demasiadas funcionalidades. Posteriormente, comprobamos que en GitHub había que crear las tareas de cada Historia de Usuario a mano, por lo que vimos que era muy costoso.

Por último, usamos *Jira*. Siguiendo algunos tutoriales, no resultaba demasiado difícil de manejar, y era una herramienta bastante útil para metodologías ágiles.

Tras probar las herramientas anteriormente mencionadas y debatir cuál sería la más adecuada, decidimos usar ***Jira***, pues a pesar de que es algo complejo de usar al principio por la cantidad de funcionalidades que ofrece, teníamos la oportunidad de usar una herramienta más completa y a la que se le suele dar uso de forma profesional.

## **b) Validación de las Historias de Usuario**

Para validar las Historias de Usuario, nos juntamos todo el grupo. Nos las repartimos de tal forma que cada subgrupo en el que nos dividimos revisara solo algunas, y si surgía alguna duda o algunas personas no lograban ponerse de acuerdo respecto a una Historia de Usuario concreta, lo debatíamos entre todos los integrantes del grupo.

Para determinar si una Historia de Usuario era válida, el equipo intentaba demostrar que ésta, cumplía cada una de las características “INVEST” (mediante ejemplos si era necesario) y de igual forma con los criterios de aceptación y el método “SMART”.

En caso de que el equipo no quedase satisfecho con una historia, ésta era reformulada hasta que todo el grupo estuviese de acuerdo.

De este modo, pudimos revisarlas todas rápidamente.

Tuvimos el caso de un par de Historias de Usuario épicas, que tuvimos que dividir en varias Historias de Usuario por separado. Resultaron ser épicas debido a que juntaban demasiadas funcionalidades en una sola Historia de Usuario. La resolución del problema fue relativamente simple; nos pusimos a debatir sobre cuál sería el mejor modo de dividirlas, hasta que quedábamos conformes con el resultado.

Otro problema que surgió fue la duda a la hora de manejar Historias de Usuario repetidas. Sin embargo, esta duda fue rápidamente resuelta por la profesora en el foro, pues solo si son exactamente iguales, se

## **C) PRIORIZACIÓN DE LAS HISTORIAS DE USUARIO**

Hemos decidido utilizar como método de priorización el método Kano porque es el que más se ajusta a la gestión de las historias de usuario que llevamos en nuestro proyecto al tener como última fase HU de tipo indiferente, cosa que no tenemos en el método MoSCoW y el método de ROI no nos parecía el método más adecuado para nuestro proyecto porque en el de Kanto está más claro y mejor explicado.

El método de los 100 puntos nos parece menos intuitivo y menos aclarativo y por eso no lo hemos utilizado.

**Obligatorias:**

* Dona quiere añadir un jugador para incluirlo en su respectivo equipo y a la lista de jugadores.
* Dona quiere añadir un equipo para incluirlo en el torneo y en la lista de equipos.
* Dona quiere añadir un entrenador para incluirlo en su respectivo equipo y a la lista de entrenadores.
* Dona quiere iniciar sesión para tener permisos especiales de administrador en la aplicación.
* Dona quiere dar de baja a cualquier jugador para eliminar jugadores de la lista de jugadores y de su respectivo equipo.
* Dona quiere dar de baja a cualquier equipo para eliminar equipos de la lista de equipos y del torneo.
* Dona quiere dar de baja a cualquier entrenador para eliminar entrenadores de la lista de entrenadores y de su respectivo equipo.
* Darnell quiere iniciar sesión para tener más funcionalidades en la aplicación.
* Dona, Darnell y Pete quieren conocer la información básica de un equipo para diferenciarlo del resto de equipos.
* Dona, Darnell y Pete quieren conocer los jugadores de un equipo para saber si está formado por buenos jugadores, y así apostar en su favor.
* Dona, Darnell y Pete quieren conocer las estadísticas de un equipo para saber toda información referente a ese equipo.
* Dona, Darnell y Pete quieren conocer las características de un entrenador para saber toda la información referente a ese entrenador.
* Dona, Darnell y Pete quieren saber los equipos en los que ha estado un entrenador para saber a cuántos equipos ha estado entrenando.
* Dona, Darnell y Pete quieren conocer las estadísticas de los equipos de un entrenador para saber si el entrenador saca el mejor rendimiento de los equipos.
* Dona, Darnell y Pete quieren ver las estadísticas de tiro de un jugador para saber si debería apostar por su equipo, o no.
* Dona, Darnell y Pete quieren ver las características de un jugador para saber si debería apostar por su equipo, o no.
* Pete quiere poder registrarme en la aplicación para tener privilegios de usuario registrado.

**Necesarias:**

* Dona quiere modificar los datos de un jugador para actualizar la información de dicho jugador.
* Dona quiere modificar los datos de un equipo para actualizar la información de dicho equipo.
* Dona quiere modificar los datos de un entrenador para actualizar la información de dicho entrenador.
* Darnell quiere rellenar un formulario del torneo para tener una base a la hora de hacer su apuesta en la web correspondiente.
* Dona, Darnell y Pete quieren poder hacer una búsqueda por nombre de equipo para poder encontrar a ese equipo más fácilmente.
* Dona, Darnell y Pete quieren poder hacer una búsqueda por nombre y apellidos de un entrenador para poder encontrarlo más rápidamente.
* Dona, Darnell y Pete quieren poder hacer una búsqueda por nombre y apellidos de un jugador para poder encontrarlo más rápidamente.

**No esperadas:**

**Indiferentes:**

* Dona quiere mover un jugador de un equipo a otro para cambiar el jugador correspondiente al equipo en el que juega actualmente.
* Dona quiere mover un entrenador de un equipo a otro para actualizar la información de los equipos y los entrenadores.

Una de las primeras cosas que hemos priorizar al nivel de ser obligatorias, ha sido la funcionalidad de la aplicación. Esto incluye la visualización de todas las estadísticas e información de los jugadores, entrenadores y equipos para, en caso de que la aplicación saliera antes al mercado, tener la funcionalidad básica ya hecha.

También hemos puesto como prioridad obligatoria el loguearse en la aplicación para permitir que los administradores puedan empezar a meter datos en la base de datos, por eso también está priorizado la funcionalidad añadir y dar de baja para poder borrar aquellos jugadores, entrenadores o equipos que ya no pertenezcan al torneo. Y dado que tenemos el login hecho hemos priorizado en permitir a un usuario registrarse en la aplicación.

Después, en una prioridad de necesarias, hemos priorizado la funcionalidad de buscar, modificar y rellenar el bracket. Todas ellas aunque son bastante importantes, no son obligatorias al empezar la aplicación. Buscar es bastante importante para darle dinamismo a la página pero no es obligatoria, ya que los jugadores se pueden encontrar no de forma rápida pero es posible. Modificar, al principio no es importante dado que es raro que un jugador, entrenador o equipo tenga que cambiar se algún dato nada mas introducirlos. Y rellenar el bracket es una funcionalidad que interesa tener, pero que en caso de falta de tiempo se podría prescindir.

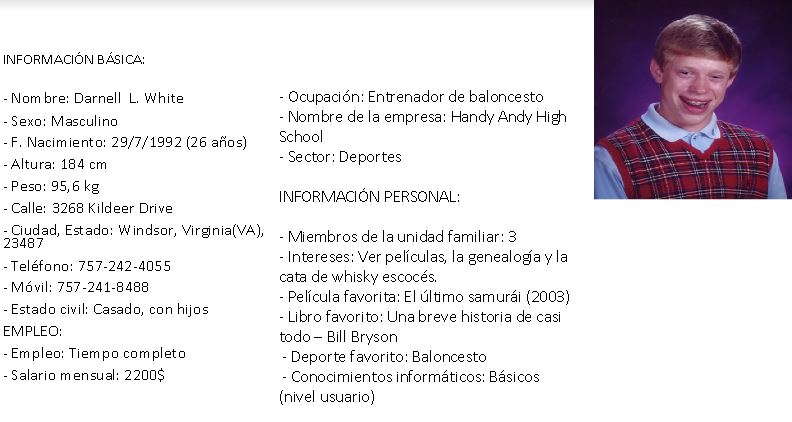
Por ahora no hemos pensado en ninguna funcionalidad no esperada, pero puede que mientras avance la aplicación pueda surgir alguna funcionalidad no contemplada.

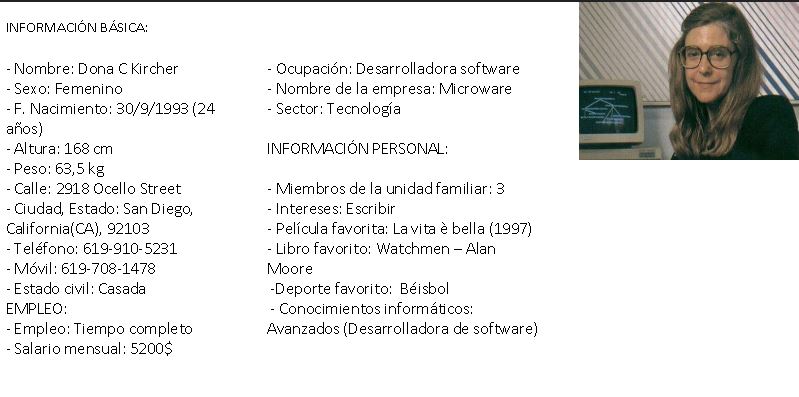
Por último, hemos dejado las funcionalidades de mover jugadores y entrenadores a indiferentes. Estas funcionalidades son útiles para mejorar la experiencia de un administrador pero no necesarias, dado que con la funcionalidad de modificar se podría realizar esta misma tarea, aunque de una forma menos rápida y dinámica.

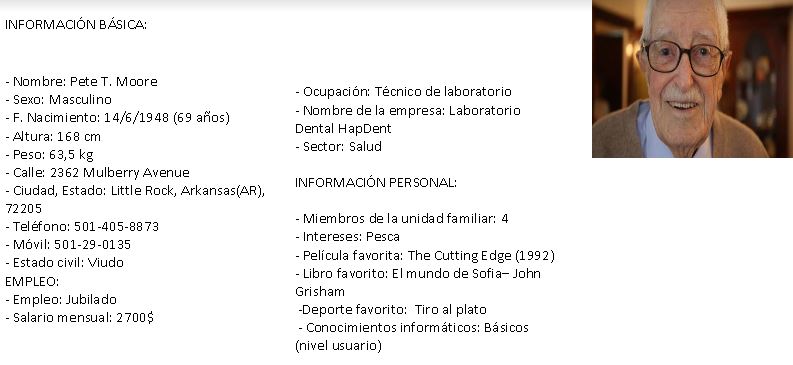
## **D) MOCKUPS CREADOS**

Para la creación de los mockups se ha optado por un diseño de interfaz simple, amigable e intuitivo, para que a un usuario no le fuera complicado el uso de la aplicación la primera vez que se encuentre con esta. Se han diseñado las vistas principales de la aplicación como son: la pantalla principal, una pantalla de log-in para los usuarios, los distintos perfiles con la información de los jugadores, equipos y entrenadores, y distintas pantallas donde mostrar las estadísticas pedidas. Todas las imágenes se encuentran incluidas en la carpeta extra del proyecto.

## **e) Perfiles de usuario**







# **3.-Prototipo**

En este apartado se explicará tanto las tecnologías probadas en el prototipo que realizamos, como su desarrollo, problemas encontrados, cómo se han resuelto, y finalmente se explicará cómo ejecutarlo y probar las funcionalidades que este ofrece.

## **HERRAMIENTAS Y TECNOLOGÍAS PROBADAS**

Además de las herramientas usadas para que el *Product Owner* gestione el *Product Backlog*, debatimos sobre qué herramientas usar para desarrollar, y para gestionar las versiones del código de proyecto, documentación, etc.

El *Product Owner* indicó que sería una aplicación de escritorio, por lo que para programar barajamos varias opciones como **Eclipse**, o **Netbeans**, ya que son plataformas para programar en Java bastante conocidas y amigables. Elegimos esta última, pues aunque ya habíamos trabajado con Eclipse anteriormente, Netbeans incorporaba el *JavaFx*, una herramienta que sirve para diseñar interfaces gráficas. En caso de elegir Eclipse, hubiéramos tenido que instalar el plugin aparte, por lo que nos decantamos por la elección más fácil a la hora de instalación (**Netbeans**), ya que esta plataforma no difiere mucho de Eclipse.

Para el servidor de la base de datos, estuvimos debatiendo sobre el uso de **Xampp** y **Wamp**, debido a que a la hora de hacer la conexión de la base de datos con el código es bastante simple, y varios integrantes del grupo lo han utilizado previamente, lo que puede a ayudar a resolver dudas en este periodo de prueba de herramientas.

Otra ventaja que aportaban estos programas es que, al unificar varios servidores, permiten elegir en la instalación qué servidor queremos usar.

Ya que el prototipo sería ejecutado desde un *.jar*, y se tendría que generar la base de datos a partir de dicho ejecutable sin depender de software externo, se propuso también el uso de **SQLite**, ya que va integrado dentro del propio proyecto, y sería sencillo realizar la conexión a la base de datos.

Debido a las restricciones de la entrega del prototipo, y habiendo probado las plataformas mencionadas, decidimos usar **SQLite** para las bases de datos. Tras acordar esto, se comentó que con **SQLiteStudio** se podría gestionar fácilmente la base de datos creada.

Tras haber probado el *SQLiteStudio*, y haber creado algunas tablas en la base de datos que usaríamos para el prototipo, nos dimos cuenta de que fallaba la inserción porque el *SQLiteStudio* nos daba problemas. Ya que este programa no era tan amigable como pensábamos, probamos también con el ***DB browser for SQLite*,** una herramienta similar.

Estuvimos probándola, y además de que no nos daba ningún error al hacer las inserciones en la base de datos (ni de otro tipo), resultó ser más intuitiva y amigable que el *SQLiteStudio*, por lo que decidimos usar al final el *DB browser for SQLite* para gestionar las tablas de la base de datos.

## **B) DESARROLLO DEL PROTOTIPO**

A la hora de crear el prototipo nos dividimos en **pequeños subgrupos** dentro de los propios desarrolladores, y nos repartimos equitativamente pequeñas partes de lo que sería el prototipo. Independientemente de la parte de código que implementara cada persona del equipo, nos aseguramos de que todo el mundo (incluidos el *Product Owner* y el *Scrum Master*, pues les podría servir para poner en práctica lo aprendido de metodologías ágiles) hubiera practicado con las herramientas que serían necesarias para el proyecto, para que el desarrollo de este fuera lo más fluido posible.

Comenzamos haciendo las interfaces gráficas que componen el prototipo, y las implementaciones de los botones que las forman, de las cuales se encargaron Carlos, y Guillermo. Hubo problemas al principio sobre todo con el manejo del repositorio de GitHub, y también con la creación de interfaces gráficas, pues era la primera vez que usábamos tanto GitHub como NetBeans. Dicha inexperiencia se solucionó consultando foros y tutoriales de internet, en los que hacen una explicación sencilla sobre cómo usar dichos programas.

Posteriormente, Marina, Jun, Sergio y Laura se encargaron de programar la parte de Controlador, así como de la creación de Factorías, de las SA, los DAOs, y de la conexión con la base de datos. Aquí volvimos a tener un problema por inexperiencia con NetBeans, que se solucionó de la misma forma que con las interfaces. Sin embargo, a la hora de hacer la conexión con la base de datos mediante el *SQLite*, (generando al principio las tablas con el *SQLite Studio)* hubo ciertas dificultades, pues, de nuevo, nos encontrábamos ante un programa que no habíamos usado anteriormente. Gracias a la colaboración de otros compañeros del equipo que habían usado el SQLite antes, y de nuevo, a tutoriales y foros de internet, pudimos realizar la conexión a la base de datos correctamente.

Posteriormente integraríamos todo. Debido a que hicimos todo el código por partes, tuvimos que integrarlo tras haber programado cada uno nuestra parte. De la integración se encargaron inicialmente tanto Diego, como Eduardo, que se encargaron de modificar, y organizar tanto el código como las clases java para que pudiera ejecutarse correctamente. No resultó ser una tarea fácil, pues cada persona programaba de una forma concreta; ponía nombres de variables, clases, eventos, etc. distintos. Por ello, se pusieron de acuerdo las personas encargadas de integrar y adaptar todo para hacerlo de la misma manera, y así unificarlo todo.

Tras haber ejecutado por primera vez todo junto, surgieron errores inesperados que después solucionaríamos. En primer lugar, se mostraba constantemente una ventana cuando no debía hacerlo. Solo tuvimos que añadir unas líneas de código para solucionarlo.

Nos dimos cuenta de que, al insertar información en la base de datos, nos saltaba un error que no entendíamos de dónde venía, pues esa parte de código estaba correctamente implementada. Después descubrimos que fue debido al *SQLite Studio*, pues este corrompía los archivos de la base de datos, impidiendo así la inserción de datos en esta. Como barajamos la posibilidad de usar el *DB browser for SQLite,* y comprobamos que también era intuitivo y amigable, como ya se ha mencionado anteriormente, acordamos usarlo para gestionar las tablas de la base de datos, en vez de usar el *SQLite Studio.* Con este nuevo programa, no nos dio ningún problema al hacer las inserciones, y las operaciones realizadas sobre la base de datos funcionaban correctamente.

Tras tener un prototipo funcional básico, Eduardo se encargó de implementar un menú login, en el que tendrías que introducir la contraseña del administrador (el usuario y contraseña estarían en la base de datos guardados) para acceder al menú principal del prototipo. Surgieron errores de los que se encargaron tanto Guillermo como Diego (entre otros, que este menú se bloqueaba si no introducías la contraseña correcta), que se solucionaron modificando código.

También se añadió un *“bracket”* compuesto por cuatro equipos, y un panel en el menú principal para mostrar los mensajes de error y confirmación. Hubo algunos problemas con el bracket, pues se bloqueaban los equipos, y queríamos que cada vez que se seleccionara uno de los dos equipos que avanzaría (pues estaban enfrentados de dos en dos), se cambiara el equipo “finalista”. Se solventó, de nuevo, modificando código.

Por último, se solucionó un error que permitía que se insertaran datos con el mismo nombre en la base de datos, en el que simplemente hubo que añadir más comprobaciones en el código.

## **C) INSTRUCCIONES DE EJECUCIÓN Y CÓMO PROBAR EL PROTOTIPO**

## 

Para ejecutar el proyecto, primero es necesario **ejecutar el .jar**.

Aparecerá la ventana de **login**. Hay que **iniciar sesión** con la contraseña *“admin”.* En el caso de introducir la contraseña errónea, simplemente saldrá un mensaje de error. Si se introduce correctamente, aparecerá el menú principal del prototipo.

Inicialmente, la **base de datos estará vacía**. Tendrán que insertarse datos introduciendo los nombres que se desea tener en la base de datos en el cuadro de texto, y pulsando el botón “Insertar”. Se mostrará un mensaje de confirmación si se ha podido insertar en el panel del menú, y si por el contrario ha habido algún error, saldrá un mensaje de inserción fallida.

Para **mostrar los datos introducidos**, solo hay que pulsar el botón “Mostrar”.

En el menú principal aparecerán también el **bracket** cuatro **equipos predefinidos** “enfrentados” de dos en dos. Cada vez que se haga click en uno de los cuatro equipos, avanzará como finalista.

Finalmente, al pulsar el **botón “Hola Mundo”** aparecerá una alerta en la que se muestra el mensaje “Hola mundo!”.

# **4.- Dinámica grupal**

Nuestro grupo desarrolló una dinámica grupal para mejorar la convivencia y trabajar mejor en equipo.

A continuación, haremos una breve descripción de la dinámica, explicaremos cómo se desarrolló, y cuáles fueron los resultados de esta.

## **A) DESCRIPCIÓN DE LA DINÁMICA**

Para esta ocasión, una vez que el grupo ha tenido la oportunidad de conocerse y trabajar juntos, se ha decidido realizar el juego “Cuenta una historia” En esta actividad, un miembro del grupo empieza con una palabra (vale cualquier tipo de palabra: sustantivo, verbo en cualquier forma, pronombre, adjetivo, etc.…) y el resto va añadiendo una palabra más a la anterior dicha, creando así una historia entre todos los participantes. Este juego se ha elegido para dar importancia a mostrar atención a todos los miembros del equipo. En esta actividad no se han centrado tantos los esfuerzos en ejercitar la memoria, sino en que todos los participantes se vieran obligados a prestar atención a las palabras de los demás para el buen desarrollo del ejercicio.

## **B) DESARROLLO**

La actividad fue realizada el día 8 de marzo en la cafetería de la facultad. Esta vez los participantes fueron quienes organizaron la actividad entre todos. Se estableció un turno para poder recitar la historia y poder añadir una palabra nueva. Se hicieron varias historias, ya que cada vez que un jugador se equivocaba, se debía empezar de nuevo.

## **C) RESULTADOS**

Tras esta actividad, el ambiente en el grupo ha mejorado. Se tiende a prestar atención a cada persona cuando da su opinión o personalidad, sin silenciarlo ni ignorarlo.