



Facultad de Ingeniería

PROYECTO DE GRADO

IMPLEMENTACIÓN DE UNA PLATAFORMA DE TRABAJO
COLABORATIVO ORIENTADA A CLUBES DE CIENCIA

Autores:

Felipe Fava Abó

José María Aguerre Oliver

Matteo Guerrieri García

Agradecimientos

Agradecemos a Robert Álvarez y Fabio Lima por los aportes brindados para la realización de este proyecto.

A Gonzalo Vilar por los aportes en diseño y experiencia de usuario, el descubrimientos del producto y el proceso de Design Sprint fueron fundamentales para la realización del mismo.

A nuestro tutor Héctor Cancela y co-tutor Gustavo Armagno por sus consejos, ayuda y dedicación durante el proyecto.

Por último, a todas las personas que estuvieron apoyándonos en cada momento de la carrera.

Abstract

Este proyecto tiene como objetivo principal la creación de una plataforma web que permita el seguimiento de las actividades y procesos comprendidos en el contexto de un Club de Ciencia, motivado por la falta de centralización y digitalización del mismo y la ayuda y facilidad que le brinda a los orientadores para fomentar el uso de la tecnología en sus ambientes de trabajo.

A partir de esto, se diseñó e implementó un sistema que permite dicha digitalización, centralización de la información de los clubes y podría brindar a futuro, una manera de compartir, colaborar a distancia y facilitar el trabajo en las instancias de Clubes de Ciencia y sus investigaciones científicas.

Índice general

1. Introducción	6
2. Antecedentes	9
2.1. Club de Ciencia [13] [14]	9
2.1.1. ¿Qué son?	9
2.1.2. ¿Quiénes lo integran?	10
2.1.3. ¿Quién puede ser orientador?	10
2.1.4. Categorías	10
2.2. Estado del arte	11
2.2.1. Clubes de Ciencia en América Latina	11
2.2.2. Plataforma de trabajo colaborativo	12
2.2.3. Comunidad online	14
3. Descubrimiento del producto	16
3.1. Motivación	16
3.2. Design Sprint	17
3.3. Remote Design Sprint	18
3.3.1. Diagrama de Afinidad	20
3.3.2. Experiencia en el Instituto de Investigaciones Biológicas Clemente Estable	22
3.3.3. Preguntas a los expertos	24
3.3.4. Caracterización del Dominio	25
3.3.5. Bosquejos	27
3.4. Resultados del proceso	28

4. Requerimientos	29
4.1. Requerimientos funcionales	29
4.1.1. Manejo de usuarios	29
4.1.2. Manejo de clubes de ciencia	29
4.1.3. Invitación a un club de ciencia	29
4.1.4. Manejo de misiones	30
4.1.5. Manejo de carpeta de campo	31
4.1.6. Manejo de póster científico	31
4.1.7. Manejo de informe de investigación	31
4.2. Requerimientos no funcionales	32
4.2.1. Concurrencia	32
4.2.2. Usabilidad	32
4.2.3. Escalabilidad	32
4.2.4. Funcionamiento en equipos del plan ceibal	33
5. Solución	34
5.1. Arquitectura	34
5.1.1. Arquitectura de tres capas	35
5.2. Modelo de datos	36
5.3. Tecnologías	38
5.3.1. Ruby	38
5.3.2. Ruby on Rails	39
5.3.3. Javascript	39
5.3.4. ReactJS	39
5.3.5. MySQL	40
5.3.6. Cloudinary	40
5.3.7. Heroku	40
5.3.8. CircleCI	41
6. Implementación	42
6.1. Implementación del Backend	42
6.2. Implementación del Frontend	45
7. Despliegue de la solución	61
7.1. Despliegue en Heroku	61
7.2. Comparación con otras plataformas [28] [19]	62

7.3. Despliegue continuo y automatizado [18]	63
8. Calidad del software	64
8.1. Revisión de código	64
8.2. Pruebas unitarias	65
8.3. Pruebas de integración <i>end-to-end</i>	66
8.4. Validación con usuarios	67
9. Gestión	70
9.1. Metodología de trabajo	70
9.1.1. Metodología ágil	70
9.1.2. Kanban	72
9.2. Comunicación	74
9.3. Herramientas	75
9.3.1. Trello	75
9.3.2. Git	75
9.3.3. GitHub	75
9.3.4. Slack	75
9.3.5. Google Drive	76
9.3.6. Google Hangouts	76
10. Conclusiones	77
10.1. Resultados y discusión	77
10.2. Trabajo futuro	78
Anexos	84
A. Diagrama de Afinidad	86
B. Instituto Clemente Estable	93
C. Crazy 8's	97
D. Historias de Usuario	100
E. Pruebas unitarias - RSpec	101
F. Pruebas E2E - Ghost Inspector	104

1. Introducción

El acceso a dispositivos digitales, así como también a internet, ha ido en constante aumento tanto en Uruguay como en el resto de los países de América Latina en los últimos años.

Desde la creación del Plan Ceibal en 2007 como un plan de inclusión e igualdad de oportunidades, el cual tiene el objetivo de apoyar con tecnología las políticas educativas uruguayas, cada niño, niña y adolescente que ingresa al sistema educativo público en todo el país accede a una computadora para su uso personal, logrando la integración de la tecnología a la educación para contribuir a mejorar su calidad e impulsar procesos de innovación social, inclusión y crecimiento personal. [31]

La educación, y con ella los docentes, tienen el desafío de preparar a los alumnos para un mundo cambiante en continuo crecimiento y globalización. Como se menciona en [10] *“Los docentes deben preparar y prepararse para el cambio, para vivir en una sociedad global donde la información se transmite y construye permanentemente, donde todos tienen acceso al mundo virtual, las redes sociales, el correo electrónico, los cursos en línea y las aulas virtuales, entre muchos otros recursos.”* Los docentes deberían entonces poder impulsar la integración tecnológica con el objetivo de acompañar desde las aulas los cambios que se vienen sucediendo.

En Uruguay existen Clubes de Ciencia, los cuales son espacios de educación no formal, en los que niños, jóvenes y adultos pueden potenciar sus ideas y su creatividad mediante la realización de un trabajo de investigación. Los participantes luego exponen su trabajo en talleres, congresos, ferias departamentales y la feria nacional.

Durante la participación en un Club de Ciencia, los integrantes llevan a cabo numerosas tareas de investigación, actividades teóricas y prácticas. Deben

elaborar un informe de investigación y un póster científico, el cual deberá ser exhibido al finalizar el proceso. También una carpeta de campo donde se almacena la documentación que, con criterio cronológico, describe el proceso de la investigación, se registran y documentan las actividades realizadas por el Club de Ciencia, desde que se conforma como tal, pasando por todas las instancias realizadas, hasta la culminación de dicha investigación.

Aplicando metodologías de desarrollo ágil de software, se implementó una plataforma de trabajo colaborativo, en donde el enfoque para la toma de decisiones en los proyectos se basa en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto.

Las metodologías ágiles se diferencian de otras para el desarrollo de software en el enfoque que le dan a las personas que hacen el trabajo y cómo trabajan juntas. Las soluciones evolucionan a través de la colaboración entre equipos multifuncionales autoorganizados que utilizan las prácticas adecuadas para su contexto.

Una de las ventajas que estas metodologías presentan es que son mas permisivas al momento de volver a atrás algunas etapas o tener que cambiar decisiones que se tomaron anteriormente. Proponen un enfoque cíclico que va mejorando la calidad de las versiones anteriores para el diseño de software, a diferencia de otros enfoques donde el desarrollo fluye secuencialmente desde el punto de inicio hasta el punto final. Cada iteración tiene su propia fase de prueba. Permite implementar pruebas de regresión cada vez que se agregan nuevas funcionalidades.

Uno de los focos del proyecto es la aplicación de estas metodologías desde la etapa de descubrimiento del producto hasta el desarrollo y las pruebas del mismo. Esto nos permitió, como se explica en el capítulo 3, cambiar el rumbo del proyecto ideado inicialmente hacia el resultado finalmente obtenido.

En conclusión, este proyecto deja plasmada la construcción de una plataforma de trabajo colaborativo orientada a Clubes de Ciencia, buscando facilitar, centralizar y brindar una mayor accesibilidad y visibilidad a sus procesos y actividades. Dentro de las funcionalidades que ofrece la plataforma se encuentran la creación del informe de investigación, la carpeta de campo y el póster científico, entre otras, las cuales son piezas fundamentales en la evaluación de un Club de Ciencia.

Al disponer dichas actividades en una plataforma de trabajo colaborativa y

buscando que los docentes orientadores integren las prácticas de enseñanza con la tecnología en los sistemas educativos, se logra favorecer el aprendizaje de los alumnos. [10]

El informe se organiza de la siguiente manera. En el capítulo 2 se presentan los antecedentes. En el capítulo 3 se explica el proceso de descubrimiento de producto llevado a cabo. En el capítulo 4 se listan y detallan los requerimientos del sistema. En el capítulo 5 se presenta el diseño de la solución del sistema. En el capítulo 6 la descripción del sistema implementado y en el capítulo 7 se detalla el despliegue de la misma. El capítulo 8 corresponde a la calidad del software, donde se presentan las pruebas realizadas y sus resultados. En el capítulo 9 se detalla la gestión del proyecto. Finalmente en el capítulo 10 se presentan las conclusiones, los resultados y trabajo a futuro.

2. Antecedentes

En este capítulo se detallan los antecedentes del proyecto presentado en este informe. En la sección 2.1 se introducen los Clubes de Ciencia, se da la definición y se describen las características de los mismos. En la sección 2.2 se detalla el estado del arte de los Clubes de Ciencia en América Latina, el relevamiento de diferentes plataformas de trabajo colaborativo y el concepto de comunidad online.

2.1. Club de Ciencia [13] [14]

El proyecto de grado se enmarca en el contexto de los Clubes de Ciencia, por lo que es importante entender y conocer más en profundidad el concepto de Club de Ciencia.

2.1.1. ¿Qué son?

El la pagina web del Ministerio de Educación y Cultura (MEC) [14] se define a un Club de Ciencia como *“un escenario de educación no formal, en el que niños, jóvenes y adultos pueden potenciar sus ideas y su creatividad a través de una investigación”*.

El Consejo de Educación Inicial y Primaria [13] también define a los Clubes de Ciencias, como un escenario de educación en el que niñas, niños, jóvenes y personas adultas pueden potenciar sus ideas y su creatividad a través de una investigación.

Los clubes se registran y pasan por diversas instancias a lo largo del año, mostrando su trabajo en talleres, congresos departamentales, ferias departamentales y la feria nacional.

2.1.2. ¿Quiénes lo integran?

Un Club de Ciencia lo integran un grupo de personas (niños, jóvenes o adultos) con una organización establecida que desarrollan actividades que contribuyen a la alfabetización científica y tecnológica del grupo y de la comunidad.

El grupo selecciona un tema de investigación que posteriormente desarrollarán con la colaboración de su orientador. Cada Club de Ciencia debe tener al menos un orientador. Es aconsejable la elección de un tema vinculado a problemas de interés local o de los propios participantes del Club de Ciencia.

2.1.3. ¿Quién puede ser orientador?

Puede ser un docente, un miembro de la comunidad o cualquier otra persona que pueda guiar al grupo en el proceso. Es una persona adulta estrechamente vinculada con el Club de Ciencia, que motiva, facilita y acompaña al grupo en sus actividades desde el comienzo de la investigación.

2.1.4. Categorías

Debido a que las edades de los participantes de un Club de Ciencia pueden ser muy diversas, los Clubes de Ciencia se clasifican en diferentes categorías, agrupando integrantes de edades similares. Estas son las siguientes:

- Abejitas - Educación Inicial.
- Colibrí - Educación Primaria Básica.
- Cardenal - Educación Primaria Superior.
- Churrinche - Educación Media Básica (1º, 2º, 3º ciclo básico) y Educación Media Rural (7º, 8º, 9º).
- Chajá - Educación Media Superior (1º, 2º, 3º bachillerato).
- Ñandú - Estudiantes de formación terciaria hasta 29 años (estudiantes de formación docente, escuelas técnicas, universitarios, etc).
- Tero - Mayores de 29 años o egresados de cualquier disciplina.

2.2. Estado del arte

2.2.1. Clubes de Ciencia en América Latina

Se utilizó el Buscador Timbó Foco¹ para hacer un relevamiento de las publicaciones existentes relativas a Clubes de Ciencia. Uno de los resultados arrojados por el buscador corresponde a un mapeo de Clubes de Ciencia en América Latina [33] llevado a cabo por dos académicas de la *Universidade Regional de Blumenau*, Santa Catarina, Brasil.

En Brasil y en otros países de Latinoamérica, existen colectivos que buscan, por medio de los Clubes de Ciencias, romper la forma tradicional de organización de los tiempos y los espacios destinados convencionalmente por las escuelas a la enseñanza de ciencias. Esto motivó a las autoras del estudio a realizar una recopilación de los distintos Clubes de Ciencia existentes en Brasil, así como también en Argentina, Uruguay, Chile, Ecuador, Perú, entre otros.

El trabajo incluyó a los Clubes de Ciencia que operan en las escuelas de los países de América Latina y cuyas actividades se publican en internet. Se tuvieron en cuenta para el estudio solamente los Clubes de Ciencia considerados activos² en redes sociales como *Facebook*, *Pinterest* y canales de *Youtube*.

El estudio pudo identificar, en el período de Setiembre de 2015 a Setiembre de 2017, el total de 278 [Ver tabla 2.1] Clubes de Ciencias en funcionamiento, con sitios y/o redes sociales activos, distribuídas en doce países de América Latina.

Para el caso particular de Uruguay, los autores mencionan que existe un número significativo de agrupaciones en contextos de educación científica y hay evidencia de que hace más de 30 años existen Clubes de Ciencias en el país. Sus acciones están vinculadas al departamento denominado *Cultura Científica*, del *Ministerio de Educación y Cultura* (MEC) [14].

Como se puede ver en [1], existen registros que en el año 2019 existían en Uruguay 1035 Clubes de Ciencia con 56.142 integrantes directos de los Clubes en todo el país.

Con el relevamiento llevado a cabo en el trabajo, los autores elaboraron *Rede*

¹<https://foco.timbo.org.uy>

²Clubos de Ciencia cuyos sitios y/o perfiles en redes sociales hayan sido actualizados en los últimos 12 meses.

Argentina	119
Bolivia	4
Brasil	77
Chile	9
Colombia	8
Ecuador	2
México	16
Panamá	1
Paraguay	1
Perú	5
Uruguay	30
Venezuela	6

Tabla 2.1: Clubes de Ciencia por país

Internacional de Clubes de Ciências (RICC), un sitio web³ que se constituyó en la Red Internacional de Clubes de Ciencias.

2.2.2. Plataforma de trabajo colaborativo

Durante el proceso de descubrimiento de producto, descrito en la sección siguiente, surgió que los Clubes de Ciencia de Uruguay no poseen ninguna plataforma en la cual mantener su trabajo, y carecen de una herramienta informática para centralizar toda la información relacionada a sus actividades.

Se descubrió que muchos registros relacionados a un Club de Ciencia son recopilados de forma rudimentaria, usando carpetas de papel con documentos, algunas veces escritas en computadora, muchas otras veces escritas a puño y letra, pero sin un estándar definido.

Esto motivó a orientar nuestro proyecto hacia la implementación de una herramienta colaborativa, donde múltiples usuarios puedan iniciar sesión, ingresar sus avances de trabajo y dejar plasmado sus descubrimientos y procesos de trabajo. Un lugar donde centralizar todas las actividades relativas al Club de Ciencia, para mejorar el acceso y visibilidad a la información por parte de los integrantes y también de los orientadores. Donde usuarios pertenecientes

³<https://www.clubesindeciencias.com>

al mismo Club de Ciencia pueden trabajar en conjunto hacia un objetivo en común.

A su vez, también se busca facilitar el trabajo del orientador, ya que éstos pueden ser orientadores de múltiples Clubes de Ciencias simultáneamente. Por lo que poseer una herramienta en la cual puedan encontrar de forma centralizada toda la información relacionada a los Clubes de Ciencia de los cuales son orientadores, les agregaría valor y facilitaría el trabajo.

Por lo tanto, primero se debe entender qué es una plataforma colaborativa. Según Wikipedia, *“una plataforma de trabajo colaborativo es un espacio virtual de trabajo, o sea, una herramienta informática (con frecuencia un sitio digital en Internet), que centraliza todas las funcionalidades ligadas a la conducción de un proyecto, la gestión de conocimientos y/o el funcionamiento de una organización, poniendo las mismas a disposición de los diferentes actores involucrados. El objetivo del trabajo colaborativo es facilitar y optimizar la comunicación entre las personas, en el marco de un trabajo o de un proyecto específico.”* [39]

La definición anterior encuadra perfectamente con la herramienta que se estaba visionando construir, ya que los clubes de ciencia deben de llevar adelante un proyecto, en un contexto de educación formal o no formal, en el cual están involucrados varios actores, en nuestro caso, los miembros del club, así como también el orientador del club.

Por lo que se realizó un relevamiento de dos plataformas colaborativas existentes. Las mismas fueron Google Docs⁴, ya que es una herramienta de uso bastante frecuente por cualquiera de nosotros, y una de las primeras que se nos ocurrió al pensar en una plataforma colaborativa. La otra herramienta analizada es una plataforma específica para clubes de ciencia en México⁵ que fue encontrada en el proceso de revisión de antecedentes.

Google Docs

Google Docs es una plataforma virtual creada por Google la cual permite crear y editar documentos de texto en tiempo real por uno o varios usuarios. Esta fue la primera plataforma que se evaluó, si bien es una plataforma muy completa en cuanto a las funcionalidades que ofrece, tiene la desventaja de que dicha plataforma es muy genérica, y no es algo específico para las necesidades

⁴<https://www.docs.google.com>

⁵<https://www.clubesciencia.mx>

de los Clubes de Ciencia.

Plataforma de Clubes de Ciencia de México

Luego de varias búsquedas en la web, buscando plataformas relativas a Clubes de Ciencia, se encontró una aplicación web relacionada a Clubes de Ciencia en México. Dicho sitio nos sirvió de inspiración para llevar a cabo el desarrollo de nuestro proyecto.

Debido a que la reglamentación de los Clubes de Ciencia varía por país, dicha plataforma solo es válida para los Clubes de Ciencia de México. Por lo tanto esto fue un desencadenante en la motivación de implementar una plataforma similar válida para Uruguay.

De forma de poder conocer más acerca de esta plataforma, y así obtener ideas de funcionalidades y una mejor experiencia de usuario para nuestra aplicación, se creó un usuario en la web y posteriormente se recorrió el sitio. Sin embargo, debido a que nuestro usuario no pertenecía a ningún Club de Ciencia, no se pudo encontrar mucha información.

Posteriormente se contactó al equipo detrás de Clubes de Ciencia de México con el objetivo de que, quizás, pudieran brindarnos alguna idea para la implementación. Sin embargo, su respuesta fue una invitación a visitar sus redes sociales y página web, cosa que ya se había realizado anteriormente, pero no fue de gran ayuda para entender el funcionamiento de la aplicación y las funcionalidades que ésta tiene. El email enviado se puede ver en el anexo [1](#).

2.2.3. Comunidad online

Uno de los conceptos que se manejaron al llevar a cabo el descubrimiento del producto fue el de comunidad online. Una comunidad online no es diferente a cualquier otra comunidad, excepto por el hecho de que no están físicamente en el mismo lugar.

Es un grupo de personas con algo en común, que podría incluir intereses, experiencias, ideales o metas compartidas. Detrás de una comunidad online o red social, hay personas reales con necesidades, dolores e intereses reales, las cuales se comunican e interactúan entre sí.

Las personas somos comunitarias por naturaleza, debido a que participamos en funciones, disciplinas, necesidades de información y profesiones propias de

una comunidad. Cuando construimos relaciones con las personas, ya sea de forma directa o indirecta, se construyen comunidades.

Lo que se busca en una comunidad online es construir relaciones personales y redes de confianza entre los participantes, reunir personas con intereses o perfiles comunes y vincular a estos grupos específicos de personas. Características que son muy importantes para la herramienta a construir. [\[30\]](#)

3. Descubrimiento del producto

La primera etapa de nuestro proyecto se basó en ejecutar un proceso de descubrimiento de producto, también conocido en inglés como *Product Discovery*. El mismo es un proceso iterativo que permite reducir la incertidumbre a la hora de crear un producto a partir de una idea, y enfocarlo a la audiencia adecuada.

En este capítulo se detalla el proceso de descubrimiento que se llevó a cabo para el producto a construir. En la sección 3.1 se detalla la motivación la cual nos impulsó a desarrollar esta plataforma. Luego en 3.2 y 3.3 se explica el *Design Sprint*, proceso utilizado para el descubrimiento del producto en sí.

En las siguientes subsecciones se detallan etapas de dicho proceso. En 3.3.1 se explica la primera herramienta utilizada, el diagrama de afinidad. Luego, en 3.3.2 se expresa nuestra experiencia en una actividad que se realizó en el Instituto Clemente Estable. Por otro lado, en 3.3.3 se detallan las preguntas a expertos que se realizaron en el proceso. Posteriormente, en 3.3.4 se clasifica el dominio de trabajo según distintas categorías, y en 3.3.5 se describe uno de los últimos ejercicios o etapas del proceso, que fueron los bosquejos.

Para finalizar, se explican los resultados alcanzados en 3.4

3.1. Motivación

Nuestro proyecto de grado se vio motivado por Microscopio Mágico [8], un proyecto de grado llevado a cabo por estudiantes de la Facultad de Ingeniería en el año 2019. Microscopio Mágico “...tiene como principal objetivo la creación de una herramienta que simule el comportamiento de un microscopio, motivado

por el impulso de la utilización de herramientas tecnológicas en la educación y por la búsqueda de acercar cada vez más a los niños a áreas científicas a una edad más temprana.”

Cuando se comenzó con el proyecto, nuestra idea inicial era darle continuidad a ese proyecto y poder integrarnos con el mismo, desarrollando una aplicación móvil la cual permitiera, con ayuda de un accesorio de bajo costo anexo a la cámara de un dispositivo móvil, utilizar el dispositivo móvil como un microscopio. Posteriormente, una vez tomadas las muestras con el dispositivo, poder subir las a la plataforma de Microscopio Mágico.

Sin embargo en el proceso la idea original cambió y nuestro rumbo viró hacia la creación de una plataforma colaborativa para Clubes de Ciencia, como se explicó en las secciones anteriores.

Robert Álvez¹, profesor de física de Salto y coordinador de los Clubes de Ciencia de dicho departamento, fue el principal impulsor de dicho cambio de foco o interés. Luego de entablar varias conversaciones con él, se fueron descubriendo las necesidades y carencias que hoy en día tienen los Clubes de Ciencias, lo cuál motivó para plantearnos el cambio de idea, y construir una herramienta entorno a los clubes.

3.2. Design Sprint

El proceso de Design Sprint² fue creado en 2010 por Jake Knapp³ en Google. Su inspiración surge principalmente en la cultura de desarrollo de productos de Google, los talleres, también de ideas como *Getting Real* de Basecamp y *The Checklist* de Atul Gawande, y su propia experiencia en la creación de productos como Gmail y Hangouts.

Desde 2010 en Google, fue perfeccionando el Design Sprint con equipos como Chrome, Search y Google X. En 2012, llevó Design Sprints a Google Ventures, donde el resto del equipo aportó su experiencia para perfeccionar el proceso.

En 2012 y 2013, el equipo de Google Ventures publicó un manual sobre Design Sprints y el proceso comenzó a hacerse conocido. El libro *Sprint* se publicó en 2016 y, en la actualidad, miles de equipos de todo el mundo han adoptado dicho proceso en startups (como Slack y Airbnb), grandes empresas

¹<https://www.twitter.com/alvezr18>

²<https://www.thesprintbook.com>

³<https://jakeknapp.com>

(como LEGO y Google), agencias (como IDEO y McKinsey), escuelas (como Stanford y Columbia), gobiernos (como el Reino Unido y la ONU) e incluso museos (como el Museo Británico y el Smithsonian).

La idea detrás del Design Sprint es construir y probar un prototipo en solo cinco días. Tomando un equipo pequeño, se avanzará rápidamente del problema a la solución utilizando una *checklist* paso a paso. Es como avanzar rápidamente hacia el futuro para que se pueda ver cómo reaccionan los clientes antes de invertir todo el tiempo y los gastos necesarios para crear un producto real. El sprint es un proceso de cinco días para responder preguntas comerciales críticas mediante el diseño, la creación de prototipos y la prueba de ideas con los clientes.

Suponiendo que el Design Sprint comienza un lunes, el proceso incluye las siguientes etapas: el lunes, se hace el mapa del problema. El martes, cada individuo esboza soluciones. El miércoles, se decide qué bocetos son más fuertes. El jueves, se construye un prototipo realista. Y el viernes, se prueba ese prototipo con cinco clientes objetivo. [26]

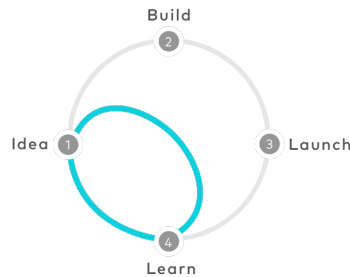


Figura 3.1: El sprint brinda a los equipos un atajo para aprender sin construir ni lanzar un producto

3.3. Remote Design Sprint

Debido a la crisis sanitaria provocada por la pandemia de COVID-19, nuestras reuniones presenciales se vieron limitadas, lo que resultó en vernos obligados a ejecutar el proceso de Design Sprint de forma remota.

Para nuestra fortuna, Jake Knapp en conjunto con John Zeratsky⁴ y Jackie Colburn⁵ crearon una guía para llevar a cabo el proceso de Design Sprint remotamente. En la guía explican como manejar videos, pizarras, votaciones, facilitación, entrevistas y todo lo necesario para llevar adelante la sesión.

⁴<https://johnzeratsky.com>

⁵<https://www.jackiecolburn.com>

Los autores mencionan en su pagina web⁶: *“Sabemos mucho sobre Design Sprints. Jake creó el Design Sprint en 2010, y juntos Jake y JZ corrieron cientos de sprints, perfeccionaron las tácticas y el proceso, escribieron el libro de Sprint más vendido y capacitaron a miles de personas para ejecutar sus propios sprints. Sin embargo, no somos expertos en sprints remotos; solo hemos participado en algunos sprints en video. Creemos que la experiencia de primera mano es fundamental, por lo que contratamos a la comunidad mundial de Design Sprint para que nos ayudara a crear esta guía. Sabíamos que este sería un proyecto gigante, por lo que le pedimos al experto facilitador de Design Sprint, Jackie Colburn, que nos ayudara a recopilar y sintetizar sus consejos. Con la ayuda de Jackie, hemos recopilado las opiniones de más de 100 personas en todo el mundo que han dirigido colectivamente miles de Design Sprints remotos.”*

La guía se estructura en 6 secciones, las mismas son:

- Preparar el escenario: El trabajo de preparación es especialmente importante para un Design Sprint remoto. Se deben elegir y sentirse cómodo con nuevas herramientas de software, y se preparan a las personas que estarán en el sprint. En esta etapa, se pone a punto todo lo necesario para el Design Sprint remoto.
- Mapeo: En esta parte del proceso, se busca realizar diagramas de afinidad, preguntar a los expertos sobre el producto y responder preguntas del estilo “¿Cómo podríamos hacer para...?”, entre otras. Esto permite al equipo recabar la máxima cantidad de información lo más rápido posible.
- Bosquejos: Esta etapa busca que los colaboradores realicen dibujos del producto a construir, para visualizar cómo funcionarían en la práctica.
- Decidir: Al llegar a esta etapa nos vamos a encontrar con muchas ideas, pero no se pueden ejecutar todas. La creación de prototipos y las pruebas son procesos que requieren mucho tiempo y recursos. Es por ello que se necesita reducir todas las ideas a una única opción sólida.
- Prototipo: En esta etapa lo que hay que hacer es convertir nuestras historias de usuario en un prototipo realista. Estos prototipos se ven bien y

⁶<https://www.thesprintbook.com/remote>

permiten a los usuarios interactuar con ellos como si fueran un producto real.

- **Testeo:** Con un prototipo diseñado, el equipo establece sesiones de prueba de usuarios. Durante estas sesiones, se presenta el prototipo a usuarios reales. Se puede probar sus suposiciones y observar cómo interactúan realmente con su solución.

En este capítulo se dio una breve introducción a las etapas del proceso. Es importante aclarar que no se realizaron todas las etapas del mismo al momento de la ejecución. La instancia de prototipado en realidad fue incluida y llevada a cabo en conjunto con el desarrollo y las pruebas fueron realizadas durante la construcción del mismo.

Se utilizó la herramienta Miro⁷ para realizar nuestro Design Sprint y el resultado se puede observar en https://miro.com/app/board/o9J_kuuyyyY=

Algunas de las herramientas o instancias de las distintas etapas del proceso realizadas nos parecieron interesantes de resaltar y se presentan en las siguientes sub-secciones. Para ver en más detalle el proceso se puede acceder a la web <https://www.thesprintbook.com/remote>.

3.3.1. Diagrama de Afinidad

Un diagrama de afinidad es una buena herramienta para organizar información desestructurada recolectada de investigaciones de usuarios, y explota el poder de la colaboración para obtener la información organizada de una manera rápida, de forma tal de construir un consenso entre el equipo durante la realización del ejercicio.

El objetivo es crear una cantidad de grupos limitada y ayudar a comprender mejor un problema. Se conocen también como el *K-J Method* ya que quién le dio el nombre fue Jiro Kawakita.

Durante el ejercicio, se muestra a todo el equipo la información cruda que se tiene sobre el problema. Luego se juntan y cada participante puede anotar todo tipo de información que crea ser relevante para mejorar el proceso a lo largo de 10 o 15 minutos. Siguiendo eso, se agrupan las ideas en colecciones que tengan sentido y describan a todas las notas contenidas en esa agrupación.

⁷<https://miro.com>

Para finalizar, se le da un nombre a cada grupo de notas y así queda definida la estructura y las áreas en donde se deberá trabajar.[4]

En resumen, los pasos del ejercicio son los siguientes:

1. Presentar el problema y brindar al equipo toda la información que se tenga.
2. En silencio, cada participante debe escribir en tarjetas o *post-its* una observación descriptiva en si misma e ir colocándolas en la pared de trabajo. (Durante 15 minutos aproximadamente)
3. Como equipo, agrupar las tarjetas en grupos afines. Se debe alentar posibles discusiones que puedan surgir y eliminar tarjetas duplicadas y darle un nombre a cada grupo.

En la figura 3.2 se puede ver nuestro diagrama de afinidad. En el Anexo se agregaron capturas de cada uno de los grupos mas detalladamente.

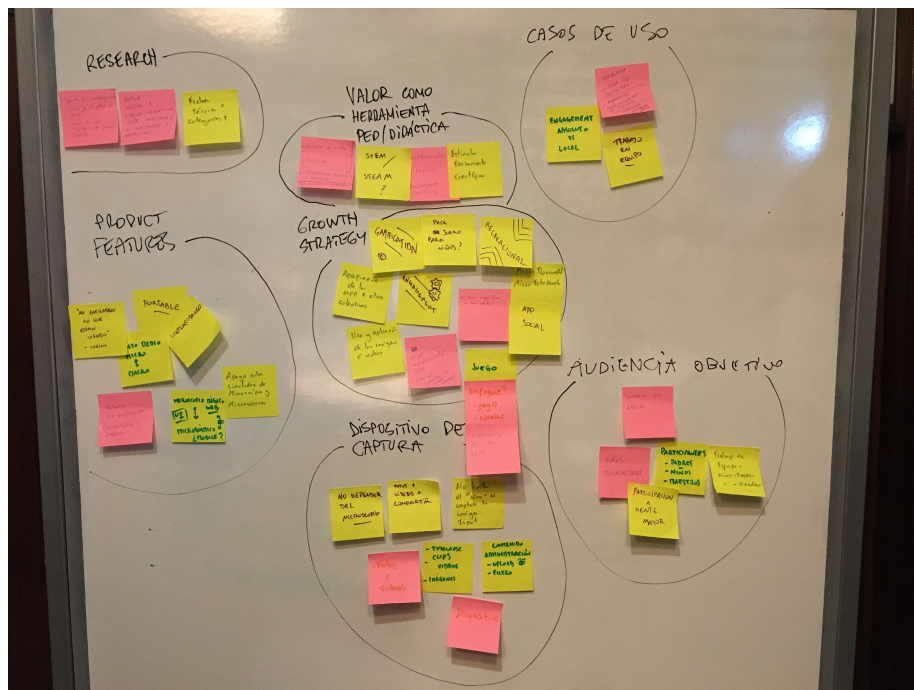


Figura 3.2: Diagrama de Afinidad resultado

3.3.2. Experiencia en el Instituto de Investigaciones Biológicas Clemente Estable

Durante el proceso de descubrimiento del producto se nos presentó la oportunidad de participar en una jornada con motivo del Día del Patrimonio⁸, en la cual se acudió al Instituto de Investigaciones Biológicas Clemente Estable⁹, en conjunto con los integrantes del proyecto de grado Microscopio Mágico, con el objetivo de validar nuestra idea inicial de realizar un producto que permita a tanto niños como adultos tomar imágenes de microscopía, con dispositivos conectados a un teléfono celular para luego poder ser compartidas digitalmente en una plataforma a crear.

Se decidió también llevar un cuestionario para realizar a modo de pequeña entrevista a los niños y adultos que se acercaran al *stand* con preguntas que nos permitieran entender mejor el producto que se pensaba construir.

Aunque resultó bastante complicado recopilar información mediante esas entrevistas a los niños, en la experiencia se los pudo ver usando el microscopio, observar sus acciones y reacciones. En la figura 3.3 se puede ver una imagen de la actividad recién mencionada.

⁸<https://www.gub.uy/ministerio-educacion-cultura/tematica/dia-del-patrimonio>

⁹<https://www.gub.uy/ministerio-educacion-cultura/iibce>



Figura 3.3: Día del Patrimonio - Instituto de Investigaciones Biológicas Clemente Estable

Lo primero que se notó fue un gran interés de los niños al observar las cosas cotidianas a gran escala en la pantalla, se sorprendían al ver como lucían cosas como la cáscara de una mandarina, la tela de la ropa que llevaban puesta, la tinta de una impresión, su propia piel, cabello, y hasta una mosca muerta.

A partir de los pocos niños a los cuales se les realizaron algunas preguntas, que fueron 8, se concluyó que la mayoría (5) de ellos había alguna vez utilizado un microscopio, aunque sea de juguete. Todos ellos eran de colegios privados y en su casa contaban con computadora, celular o tablet, la cual o bien era de ellos o de sus padres. Debido a que eran de colegios privados, ninguno tenía ceibalita. La edad de los entrevistados varió entre 5 y 10 años, el promedio fue 8 años. En [B.1](#) se pueden ver los resultados obtenidos luego de las entrevistas.

También se pudo observar que además de los niños, los padres de estos se mostraron interesados con los microscopios y los utilizaron para observar distintas cosas del lugar. Por otro lado, se presentó un grupo de maestras al *stand* a probar los microscopios, y tuvieron una reacción muy positiva. Se les

comentó la idea del proyecto, que podría estar enfocado al ámbito escolar, y se mostraron interesadas.

Se pueden observar más imágenes [B.2](#), [B.3](#) y [B.4](#) de esta experiencia en el anexo.

3.3.3. Preguntas a los expertos

Como se mencionó previamente, se consultó con algunos expertos en la materia durante el proceso.

Uno de ellos fue Robert Álvez. Con él se tuvo una fructífera entrevista, la cual se puede encontrar completa en [16]. En la misma, se pudo hablar de qué es un Club de Ciencia, cómo funcionan, qué son las ferias departamentales y nacionales y todo tipo de aspectos relacionados a las actividades y desarrollo de un Club de Ciencia. A partir de las conversaciones con Robert, fue cuando se decidió enfocar el proyecto a los Clubes de Ciencia en sí. Toda esta información nos ayudó a entender de mejor manera el contexto para el cual se crearía la plataforma.

A continuación se puede ver un fragmento de la entrevista, a partir del minuto 47, el cual es el momento clave que posteriormente nos haría cambiar el objetivo de nuestro proyecto:

[...]

- Gustavo: [...] el valor agregado no está tanto en el dispositivo, ¿no?, en realidad...

- José: No... sin duda el valor agregado está en la aplicación, es decir, en lo que haga la aplicación. Como conecte la aplicación a estos distintos usuarios, digamos, los docentes con los alumnos investigadores, con el Club de Ciencia o los “coordinadores” del club.

- Robert: ¡Orientadores!

- José: ¡Orientadores, exacto!

[...]

- Gustavo: Como que manualmente en realidad, se podría hacer. [...] Suben a Google Drive las fotos que van sacando, o a Dropbox, o lo que sea, se va armando la ficha de investigación. Estoy tratando de enfocarme en la necesidad real, ¿no? Porque ahí hay un caso, muchas veces las aplicaciones lo que tienen

es un factor de, esto, de lo lúdico, te engancha por otro lado, o te resuelve algún workflow, que en realidad es medio tedioso de realizar manualmente.

[...] (En este momento, nos ponemos a conversar sobre como funcionan los centros médicos, específicamente en el rubro de imagenología, donde los sistemas facilitan el trabajo, definiendo un *workflow* a la hora de hacer un trabajo de este tipo)

- José: En realidad lo que termina haciendo es guiándote a llenar la ficha, a incentivarlo con logros, con “achievements”, con metas, con puntos.

- Gustavo: Claro, pero ahí está el ajuste fino.

- José: Ahí está el ajuste fino de para qué va a funcionar.

- Gustavo: Para mí hay que estudiar más el tema de los Clubes de Ciencia, ver [...] alguna información vinculada a como se da todo este proceso manualmente, porque capaz que ahí podemos facilitar.

- Robert: ¡Ajá!

- José: Meternos en un parte de ese proceso y solucionar eso.

- Gustavo: Claro, exacto.

- Robert: Claro.

[...]

- Gustavo: Capaz que ahí hay algo que no estamos viendo. [...] Toda esta conexión en red, de los distintos clubes, capaz que hay algo más grande que no estamos viendo, ¿no? Que es facilitar, ofrecer una plataforma para que se conecten los distintos clubes...

- Robert: Claro, una base de todo.

[...]

Una vez decidido el cambio de foco sobre la herramienta a construir, surgió la oportunidad de hacerle algunas preguntas a Fabio Lima, un compañero de trabajo, el cual fue profesor de Física en Montevideo y orientador de varios Clubes de Ciencia. Dicha entrevista fue de gran ayuda a lo largo del proceso.

3.3.4. Caracterización del Dominio

Del reglamento de las ferias estandarizadas publicado por el Ministerio de Educación y Cultura [15] y de la entrevista realizada a Robert Álvarez [16] se obtuvo la información necesaria para realizar el análisis del dominio, el cual se dividió en categorías según los diferentes actores y elementos.

- Por rol
 - Coordinador
 - Orientador
 - Participantes (estudiantes)
 - Jurado
- Por evento
 - Talleres con estudiantes
 - Talleres docentes
 - Espacios de coordinación docente
 - Semana de Ciencia y Tecnología
 - Feria departamental
 - Feria nacional
 - Congresos departamentales de Clubes de Ciencia
 - Talleres regionales para orientadores
 - Mini-congreso virtual
 - Campamento científico
- Por intereses temáticos
 - Interés local
 - Interés regional
 - Interés individual de cada club
- Por nivel formativo
 - Abejitas
 - Colibrí
 - Cardenal
 - Churrinche
 - Chajá
 - Ñandú

- Hornero
- Por ubicación geográfica
 - Departamental
 - Nacional
 - Internacional
- Por área de conocimiento
 - Social
 - Tecnológica
 - Científica
- Por institución
 - Ministerio de Educación y Cultura (MEC)
 - Escuela Pública
 - Liceo Público
 - Colegio
 - Otros (INAU, CAIF, Centros MEC, Casa de Cultura, etc.)

3.3.5. Bosquejos

Esta es una actividad fuera de línea para el equipo. El equipo debería trabajar en papel y no es necesario realizar la actividad en conjunto.

Para realizar esta etapa del proceso se llevó a cabo una técnica de diseño llamada *Crazy 8's*¹⁰, cada integrante tomó una hoja A4 y la dividió en 8 partes iguales, y la consigna era dibujar en el tiempo de 8 minutos, diferentes pantallas que se nos vinieran a la cabeza del producto.

Es un ejercicio de dibujo rápido que desafía a las personas a esbozar ocho ideas distintas en tiempo acotado. En las figuras C.1 y D.1 se puede ver dos ejemplos del resultado que se obtuvo luego de la ejecución del proceso.

¹⁰<https://www.youtube.com/watch?v=yz4g87XapQ0>

3.4. Resultados del proceso

A lo largo de este proceso se fue construyendo la idea en la que finalmente se enfocó el proyecto. Al principio, como se explicó anteriormente, la idea era crear una aplicación que permitiera, incluyendo un dispositivo de bajo costo, tomar imágenes cercanas a la microscopía y compartir dichas imágenes con la comunidad a través de la plataforma Microscopio Mágico.

A medida que se fue avanzando con el proceso de descubrimiento, comenzaron a aparecer indicios de la existencia de una necesidad dentro del contexto de los Clubes de Ciencia, la cual nos pareció mucho mayor a la necesidad sobre una herramienta de microscopía de bajo costo.

Alrededor del minuto 47 de la entrevista realizada a Robert Álvarez [16], surgió la interrogante de cuál era realmente el valor agregado que se le podría brindar a los Clubes de Ciencia, alumnos y orientadores, y si no sería mejor aprovechada una plataforma que abarcara mas ampliamente las actividades pertinentes a los Clubes de Ciencia.

Es por ello que se decidió descartar el desarrollo de la aplicación inicial ya que se consideró que la misma era fácilmente sustituible. Se notó que la construcción de un dispositivo de microscopía para el dispositivo móvil¹¹, en conjunto con herramientas online existentes para compartir imágenes, era fácilmente accesible y de bajo costo. Por lo que finalmente se decidió enfocar nuestros esfuerzos en la creación de una plataforma de trabajo colaborativo orientada a Clubes de Ciencia.

Esto nos permitió seguir con las distintas etapas del descubrimiento del producto ya con otra idea en nuestras mentes, lo cual sin lugar a dudas iba a aportar más funcionalidades. De esta forma, se decidió cambiar el rumbo de lo que se iba a construir, siempre enfocados en generar la mayor cantidad de valor posible para los usuarios.

El resto de las actividades realizadas nos ayudaron a definir los requerimientos necesarios para la construcción del producto final, concentrándonos en distintos aspectos que todo desarrollo de software debe contener, poniendo énfasis en la usabilidad y siguiendo buenas prácticas de desarrollo, lo cual por lo general lleva a un producto con una buena calidad y que cumple con las expectativas deseadas.

¹¹<https://www.instructables.com/10-Smartphone-to-digital-microscope-conversion/>

4. Requerimientos

En este capítulo se presentan los requerimientos tanto funcionales como no funcionales del producto a construir. En la sección 4.1 se presentan los requerimientos funcionales y en la sección 4.2 se presentan los no funcionales.

4.1. Requerimientos funcionales

A continuación se listan los requerimientos funcionales del sistema a implementar.

4.1.1. Manejo de usuarios

El sistema permitirá el manejo del perfil de cada usuario, permitiendo que el usuario ingrese su información y ésta quede persistida en el mismo. El usuario podrá ver y modificar dicha información desde una pantalla específica.

4.1.2. Manejo de clubes de ciencia

El sistema permitirá listar, crear, editar y borrar clubes de ciencia. En el listado solo se verán los clubes en los cuales el usuario del es miembro. Cualquier usuario podrá crear un club, quedando asignado como orientador al momento de la creación. Las operaciones de borrado y edición solo las podrán realizar los orientadores.

4.1.3. Invitación a un club de ciencia

El sistema deberá permitir crear una invitación a un Club de Ciencia ingresando el email del usuario a invitar. Se creara una asociación con el club y el

usuario invitado si la invitación es aceptada. Dicha invitación también puede ser rechazada.

A continuación, en la figura 4.1 se puede ver un wireframe de dicho requerimiento.



Figura 4.1: Invitación a un Club de Ciencia

4.1.4. Manejo de misiones

El sistema permitirá listar, crear, editar y borrar misiones. La creación, edición y borrado solo podrá ser llevada a cabo por los usuarios orientadores. El listado de misiones estará disponible para todos los usuarios. Los miembros de un club podrán asignarse misiones y completarlas.

En la siguiente imagen, figura 4.2, se puede ver un wireframe de dicho requerimiento.

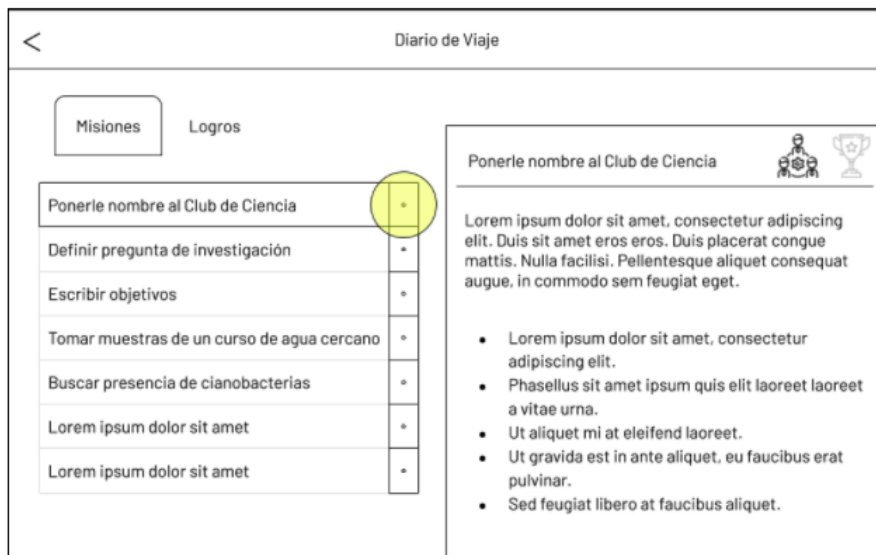


Figura 4.2: Listado de misiones de un Club de Ciencia

4.1.5. Manejo de carpeta de campo

El sistema permitirá las operaciones de listado, creación, edición y borrado de entradas de la carpeta de campo para cualquier usuario del sistema. Dichas entradas serán listadas por orden cronológico según la fecha ingresada en la misma.

4.1.6. Manejo de póster científico

Se podrá crear y editar un póster científico con secciones pre-configuradas por el sistema. También se podrá imprimir el póster para su presentación.

4.1.7. Manejo de informe de investigación

El sistema permitirá subir un documento de texto el cual será el informe de investigación del club de ciencia.

En la figura 4.3 se puede ver un wireframe de dicho requerimiento.

<

Informe de Investigación

PDF

Nombre del archivo:

informe_cianobacterias.pdf

Tamaño:

373 kb

Última actualización:

11/08/2020

URL para compartir:

clubdeciencia.org/1/h2m_hJczee0

Estado:

NO ENVIADO

Copiar al Portapapeles

Antes de enviar el Informe de Investigación para ser evaluado por el Comité de Evaluación, asegúrate de que cumple con la guía descrita más abajo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis sit amet eros eros. Duis placerat congue mattis. Nulla facilisi. Pellentesque aliquet consequat augue, in commodo sem feugiat eget.

Enviar

4.2. Requerimientos no funcionales

A continuación se listan los requerimientos no funcionales del sistema a implementar.

4.2.1. Concurrency

El sistema debe soportar el uso concurrente de varios usuarios.

4.2.2. Usabilidad

Debido a que la aplicación va a ser utilizada por niños, el sistema deberá presentar una interfaz de usuario sencilla para que sea de fácil manejo para los usuarios. Deberá ser auto-explicativo de forma de que pueda ser utilizado y comprendido fácilmente por ellos.

4.2.3. Escalabilidad

El sistema deberá ser escalable fácilmente, en caso de querer hacer crecer la aplicación al incorporar a futuro nuevas funcionalidades y en caso de que crezca

la cantidad de usuarios.

4.2.4. Funcionamiento en equipos del plan ceibal

Debido a que existe la posibilidad de que sea utilizado en un contexto escolar, el sistema deberá poder ser ejecutado en computadoras brindadas a las escuelas por plan ceibal.

5. Solución

En este capítulo se presenta la solución a construir. Primeramente en la sección 5.1 se presenta la arquitectura del sistema de forma global, en 5.2 se detalla el modelo de datos a construir, y por último, en la sección 5.3 se describen las tecnologías utilizadas para la construcción de la solución, mencionando la motivación en la elección de las mismas.

5.1. Arquitectura

El concepto de arquitectura de software refiere a la estructuración de un sistema a construir. Representa un diseño de alto nivel del sistema que tiene como objetivo servir como guía para el desarrollo de software, y satisfacer los atributos de calidad del sistema, como pueden ser el desempeño, seguridad, completitud, flexibilidad, robustez, etc.

Las decisiones críticas relativas al diseño general de un sistema de software deben de hacerse desde un principio, idealmente, se debe diseñar la arquitectura en etapas tempranas del desarrollo. El demorar u omitir esta etapa puede causar que el producto final no satisfaga adecuadamente las necesidades de los clientes y los requerimientos no funcionales del mismo.

Además, es importante tener en cuenta que el costo de las correcciones relacionadas con problemas en la arquitectura es muy elevado. Por lo que la arquitectura de software juega un papel fundamental dentro del desarrollo de un sistema. [9]

5.1.1. Arquitectura de tres capas

La arquitectura de tres capas, también conocida como arquitectura de tres niveles, divide la aplicación en tres capas lógicas distintas, donde cada una de ellas contiene grupo de interfaces bien definidas, y pueden estar físicamente distribuidas.

Un aspecto fundamental de esta arquitectura es que los componentes de una capa sólo pueden hacer referencia a componentes de capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma tal que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Como resultado, se logra una mejor mantenibilidad del sistema en general, y mayor facilidad de realizar cambios o agregar nuevas funcionalidades.

Las tres capas que define esta arquitectura son:

- **Presentación:** Es la que se encarga de que el sistema interactúe con el usuario y viceversa, muestra el sistema al usuario, le presenta la información y obtiene la información del usuario. Su función es pasarle las acciones que realice el usuario a la capa de negocio.
- **Negocio:** En esta capa se gestiona la lógica de la aplicación. Es donde se reciben las peticiones del usuario, se procesa la información y se envían las respuestas tras el proceso. Se denomina capa de negocio o capa lógica, porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar o recuperar datos del mismo.
- **Persistencia:** o también denominada capa de acceso a datos, es la encargada de almacenar los datos del sistema. Su función es almacenar y devolver datos a la capa de negocio. Aquí se gestiona todo lo relativo a la base de datos y a la creación, edición y borrado de datos de la misma. Por lo general, está formada por uno o varios sistemas gestores de bases de datos, localizados en un mismo servidor o en varios.[6]

Para la solución del sistema ha construir se decidió utilizar esta arquitectura. Una razón clave de esta decisión es por la importancia de mantener independen-

cia entre los distintos componentes del sistema, permitiendo en un futuro que otras aplicaciones puedan utilizar los servicios que brinda la capa de negocio.

Un ejemplo claro puede ser la posibilidad de crear una aplicación móvil que se integre con la capa de servicios y brinde las mismas funcionalidades que la aplicación web.

Otra ventaja importante que ofrece es respecto a la escalabilidad del sistema, ya que una arquitectura así permite escalar las capas de manera independiente según las necesidades.

Por otro lado, este tipo de arquitecturas suelen ser más fáciles para agregar nuevas funcionalidades, debido a la independencia entre capas. También, pueden ser administrados y desplegados en forma autónoma, sin afectar a las distintas capas.

A continuación, en la imagen 5.1 se presenta de forma gráfica esta arquitectura.

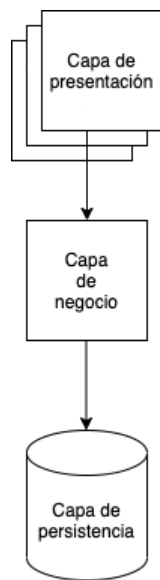


Figura 5.1: Arquitectura de 3 capas

5.2. Modelo de datos

Para cumplir los requerimientos mencionados en el capítulo anterior, se realizó un Modelo Entidad Relación (MER) (Figura 5.2) a partir del cual se

obtuvo el modelo relacional y sus dependencias.

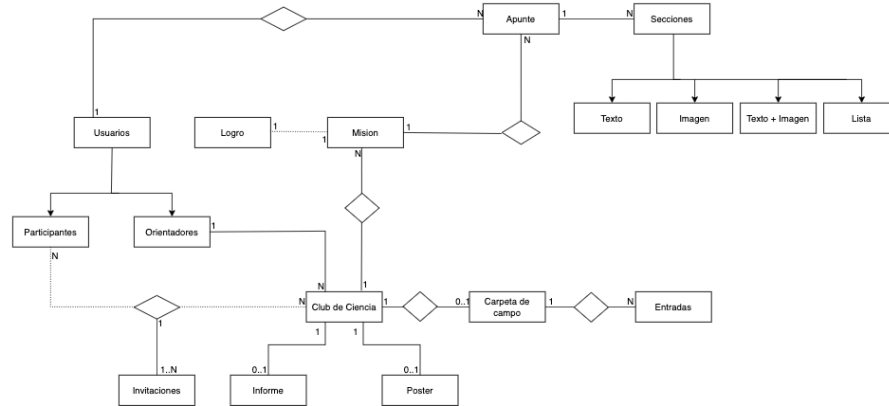


Figura 5.2: Modelo Entidad Relación

A continuación se listan las tablas que se obtuvieron para cumplir con el modelo planteado:

- users(id, email, encrypted_password, names, last_names, birthday, institution, about_me, phone, department)
- clubs(id, name, description, category, area, formal)
- roles(id, name)
- clubs_users_roles(id, club_id, user_id, role_id)
- reports(id, club_id, name, file)
- posters(id, club_id, title, abstract, results, conclusions, introduction, methodology, bibliography, acknowledgments)
- field_folders(id, club_id)
- entries(id, field_folder_id, title, description, date)
- missions(id, club_id, name, description, status)
- Logros(id, mission_id, trophy)
- notes(id, user_mission_id)

- `note_sections(id, note_id, section_type, url, text, list, position)`
- `missions_assigned_users(id, user_id, mission_id)`
- `invitations(id, status, email, club_id)`

5.3. Tecnologías

En las próximas subsecciones se presentan y describen las tecnologías utilizadas para lograr la solución detallada en la sección 6.

Como se podrá ver más adelante, para la implementación del servidor se optó por utilizar la tecnología Ruby on Rails (RoR), y para el cliente ReactJS.

Se decidió utilizar RoR para el servidor ya que cuenta con una curva de aprendizaje muy baja, es una tecnología muy popular, que cuenta con una gran comunidad activa y con una gran diversidad de librerías. Además, tiene una muy buena documentación, y por ultimo pero no menos importante, por la experiencia previa de los desarrolladores.

Por otro lado, se decidió utilizar ReactJS para el cliente principalmente por la performance, traducándose en una mejor experiencia de usuario. Adicionalmente, ReactJS es muy popular, cuenta con el respaldo de Facebook y posee una gran comunidad, con una enorme cantidad de librerías y documentación. En [5] se pueden ver en más profundidad las ventajas del desarrollo web utilizando ReactJS.

A continuación, se detallan brevemente qué son y como surgieron dichas tecnologías, además del resto de tecnologías utilizadas.

5.3.1. Ruby

Ruby¹ es un lenguaje de programación publicado en 1995 por un programador japonés llamado Yukihiro “Matz” Matsumoto. Es un lenguaje interpretado, dinámicamente tipado, orientado a objetos y su implementación oficial es distribuida bajo una licencia de software libre. [3]

¹<https://www.ruby-lang.org>

5.3.2. Ruby on Rails

Ruby on Rails², o también conocido simplemente como Rails, es un *framework*³ escrito sobre el lenguaje de programación Ruby, creado en 2004 por el programador danés David Heinemeier Hasson.

Rails permite crear aplicaciones web de forma fácil y rápida. El principal principio de este *framework* es “Convención sobre Configuración”, esto significa que Rails hace algunas suposiciones sobre lo que quieres hacer y cómo vas a hacerlo, haciendo que se pueda tener un proyecto andando de forma muy rápida sin necesidad de especificar o configurar prácticamente nada.

Entre aquellas empresas más populares que utilizan Rails se encuentran: Basecamp, HEY, GitHub, Shopify, Airbnb, Twitch, SoundCloud, Hulu, entre otras. [23]

5.3.3. Javascript

JavaScript⁴ fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript.

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. [37]

5.3.4. ReactJS

ReactJS⁵ es una librería JavaScript de código abierto enfocada a la visualización. Esta tecnología permite el desarrollo de interfaces de usuario de forma sencilla, esto es posible mediante componentes interactivos y reutilizables.

Fue desarrollada inicialmente por Facebook en marzo de 2013 para poder implementar el *timeline* de dicha aplicación de manera óptima y rápida. [7]

Actualmente es software libre y a partir de su liberación ha acaparado una gran comunidad de desarrolladores y entusiastas.

²<https://rubyonrails.org>

³<https://es.wikipedia.org/wiki/Framework>

⁴<https://www.javascript.com>

⁵<https://reactjs.org>

5.3.5. MySQL

MySQL⁶ es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation. Está considerada como la base de datos de código abierto más popular del mundo.

Fue creada originalmente por una compañía sueca llamada MySQL AB en 1994. Luego, en 2008 la empresa de tecnología de los Estados Unidos Sun Microsystems compró MySQL AB, y finalmente, en 2010, Oracle adquirió Sun Microsystems en el 2010. MySQL ha sido propiedad de Oracle desde entonces. [38]

5.3.6. Cloudinary

Cloudinary⁷ es una compañía de tecnología SaaS (Software as a Service) que ofrece servicios de gestión de imagen y video en la nube. Permite cargar, almacenar, administrar, manipular y entregar imágenes y videos para sitios web y aplicaciones. [11]

5.3.7. Heroku

Heroku⁸ es uno de los PaaS (Platform as a Service) en la nube más utilizados en la actualidad. Ofrece servicios de servidores y redes administrados en donde se pueden alojar, configurar, administrar y escalar aplicaciones de diferentes lenguajes de programación. Heroku nació en el 2007 y fue adquirida en el 2010 por Salesforce.

Además, Heroku admite más de 175 *add-ons*, que cubren varios propósitos, desde notificaciones, manejo de base de datos y hasta seguridad. [21]

En la sección 7.1 se explica por qué se eligió esta plataforma para el despliegue de la solución.

⁶<https://www.mysql.com>

⁷<https://cloudinary.com>

⁸<https://www.heroku.com>

5.3.8. CircleCI

CircleCI⁹ es una empresa fundada en 2011 y con sede en San Francisco, Estados Unidos, la cual permite automatizar los procesos de creación, prueba e implementación.

Es decir, consiste en hacer integraciones automáticas de un proyecto lo más a menudo posible para así poder detectar fallos cuanto antes.

En nuestro caso, se ejecutaban los tests automatizados del proyecto cada vez que se agregaba nuevo código, y de esta forma poder minimizar la introducción de errores al proyecto. [22]

⁹<https://circleci.com>

6. Implementación

Para la implementación se tuvo en cuenta el diseño del capítulo anterior, y los requerimientos mencionados en el capítulo 4.

La implementación se divide principalmente en dos grandes partes, por un lado el *Backend*, y por otro lado el *Frontend*. La primera es la encargada de manejar y ejecutar toda la lógica de negocio de la aplicación, es una API REST desarrollada en Ruby on Rails. Mientras que la segunda es la encargada de convertir los datos provenientes del Backend en una interfaz gráfica de forma que el usuario pueda ver e interactuar con la información de manera digital. El Frontend es una *Single Page Application* (SPA) desarrollada en ReactJS.

Para ambas implementaciones se modularizó el código de forma que pueda ser reutilizable en caso de ser necesario, más fácil de mantener y entender, y se siguió la guía de buenas prácticas de cada lenguaje usados en el Backend y en el Frontend, Ruby y JavaScript respectivamente.

Este capítulo se estructura de la siguiente forma, en la sección 6.1 se detalla la implementación del Backend con los servicios que contiene, y en la 6.2 se describe la implementación del Frontend.

6.1. Implementación del Backend

Como ya se mencionó anteriormente, el Backend es una API REST implementada en Ruby on Rails que expone servicios de forma tal que puedan ser consumidos por el Frontend. Es la encargada de implementar los servicios y la capa de acceso a datos.

Entre los principales servicios que se implementan según su funcionalidad se encuentran:

- Autenticación

Este servicio se encarga de autenticar a un usuario. Para esto, se decidió utilizar JSON Web Token (JWT)¹.

Al iniciar sesión un usuario a la aplicación, ingresando su mail y contraseña, se devuelve en el *header* ‘*Authorization*’ de la *response* un *token* (JWT), el cual el usuario podrá enviar, en el *header* ‘*Authorization*’ de las siguientes *requests*, permitiéndole solicitar recursos al servidor que requieran autenticación.

Dicho token identifica al usuario *logueado*, y esta firmado digitalmente por el servidor utilizando una clave privada, de forma tal que no pueda ser alterado por nadie que no conozca dicha clave.

- Usuarios

Este servicio permite crear, editar e iniciar sesión de los usuarios del sistema. También permite agregar y eliminar roles a un usuario existente.

- Clubes

El servicio de clubes permite listar, crear, editar y eliminar Clubes de Ciencia. También permite obtener información de un club en particular, y poder abandonar un club del cual un usuario es miembro.

- Carpeta de campo

En este servicio se pueden listar las entradas de una carpeta de campo para un club, y también crear, editar o eliminar una entrada de dicha carpeta.

- Misiones

El servicio de misiones permite crear, editar y eliminar misiones.

- Notas

Este servicio permite crear una nota asociada a una misión activa, así como también listar, editar y borrar notas.

¹JSON Web Token es un estándar abierto basado en JSON propuesto por IETF para la creación de tokens de acceso que permiten la propagación de identidad y privilegios.

- Invitaciones

Dicho servicio permite mandar una invitación a un usuario para ingresar a un club de ciencia. Manda un email al usuario al cual se esta invitando a formar parte del club, el cual puede existir en el sistema como no. En caso de no existir se lo invita a crear una cuenta en la aplicación, y al ingresar a la misma ya tendrá la invitación al club. En este servicio también se puede obtener la lista de invitaciones pendientes de un usuario, así como también aceptar o rechazar invitaciones.

- Posters

El servicio de posters permite obtener información de un poster de cierto club, así como también crear, editar y eliminar.

- Informes

El servicio de informes permite subir un archivo a Cloudinary, en nuestro caso de uso se correspondería al informe de cierto Club de Ciencia. También permite obtener la URL a dicho archivo en caso de que el Club de Ciencia tenga un informe asociado.

Por otro lado, además de los recursos mencionados, los cuales exponen *end-points*² que son consumidos por el Frontend para obtener y manipular los datos que se muestran en pantalla, el Backend cuenta con dos componentes adicionales importantes.

Por un lado, se cuenta con un módulo encargado del manejo de archivos, de forma de poder subir, almacenar y manipular archivos en la nube, que como ya se explicó en la sección 5.3, se utilizó el servicio de Cloudinary para dicho fin.

Y, el otro componente importante, es el encargado de envío de e-mails. Este módulo es utilizado para las invitaciones de un usuario a un club. Es importante aclarar que dichos envíos se hacen en segundo-plano, principalmente por *perfomance*, ya que no es necesario que el cliente espere por la ejecución de esta tarea.

²<https://rapidapi.com/blog/api-glossary/endpoint/>

6.2. Implementación del Frontend

Como se mencionó anteriormente, el Frontend fue desarrollado en ReactJS como una SPA, esto significa que la aplicación web utiliza una única página, por lo que el HTML, CSS y JavaScript se cargan una única vez.

Lo que se logra con este tipo de aplicaciones es dar una experiencia más fluida y rápida a los usuarios, ya que la página no tiene que volver a cargarse en ningún punto del proceso, y tampoco es necesario transferir a otra página, aunque sí existe la navegabilidad en páginas lógicas dentro de la aplicación.

Al tratarse de una aplicación de tamaño mediano con interacción del usuario, no se precarga toda la información de una sola vez, en cambio lo que se hace es ir cargando el contenido dinámicamente a medida que se requiere. Dicho contenido se solicita mediante llamadas AJAX³ al servidor, es decir al Backend.

Es importante aclarar nuevamente que todo esto se logra sin la necesidad de cargar toda la página de nuevo, sino que solo se actualiza el nuevo contenido mediante JavaScript.

A continuación se listan las principales vistas implementadas en el Frontend:

- Log In

Esta vista permite a un usuario iniciar sesión en la aplicación. Contiene un formulario que permite ingresar un e-mail y contraseña de un usuario, y lo valida contra el servidor. En caso que los datos sean correctos, se permite el acceso al usuario, se guarda el token JWT devuelto por el Backend para futuras solicitudes, y se hace una redirección a la vista ‘Dashboard’, la cual es explicada más adelante. En caso de error, se muestra un mensaje descriptivo del mismo.

A continuación, en la figura 6.1 se puede observar dicha vista.

³<https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>

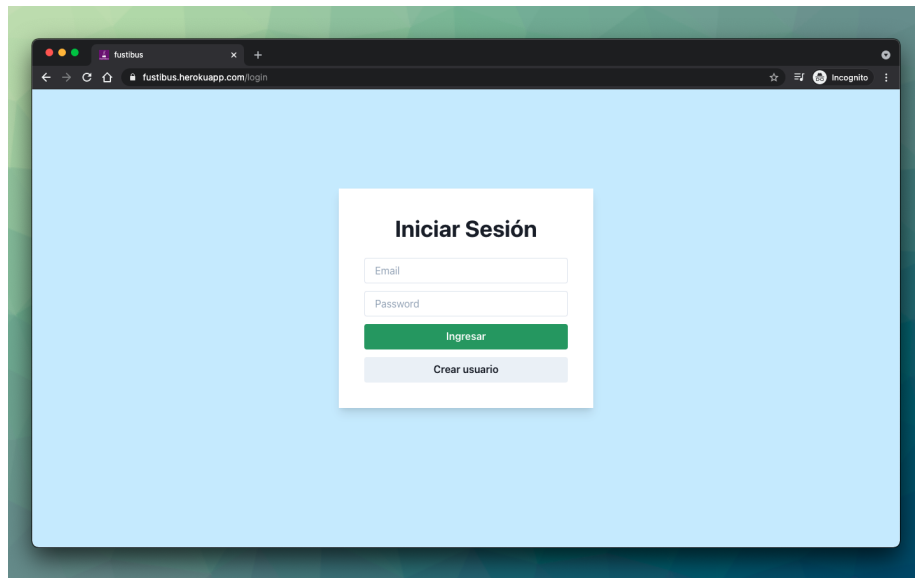


Figura 6.1: Log In

■ Sign Up

Permite crear un nuevo usuario en el sistema. Contiene un formulario que acepta un e-mail, una contraseña y una confirmación de contraseña. Con los datos mencionados, se hace el pedido al Backend para la creación del usuario. En caso de que la respuesta sea exitosa, se inicia sesión al usuario recién creado y se hace una redirección a la vista 'Welcome'. En caso de algún de error, se le muestra una descripción del mismo en pantalla.

A continuación, en la figura 6.2 se puede observar dicha vista.

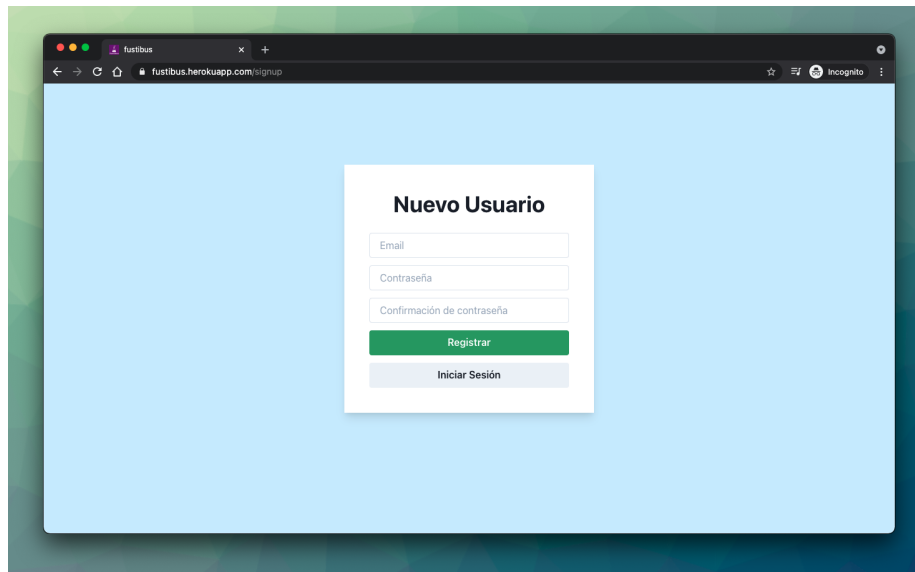


Figura 6.2: Sign Up

■ Welcome

Esta vista se muestra una única vez al usuario, cuando éste ingresa por primera vez a la plataforma. Está compuesta por 3 etapas o vistas intermedias. La primera, es un mensaje de bienvenida al nuevo usuario. La segunda, es un formulario donde se le solicita que ingrese más datos como el, o los, nombres del usuario, los apellidos, el departamento en el cual reside, y su institución. Luego, por último, se le da la opción al nuevo usuario a subir una imagen de perfil. Cabe mencionar que este último paso es opcional para el usuario, es decir, puede decidir no subir una imagen. Una vez terminado con las etapas descritas, se hace una redirección al ‘Dashboard’.

A continuación, en la figura 6.3 se puede observar dicha vista.

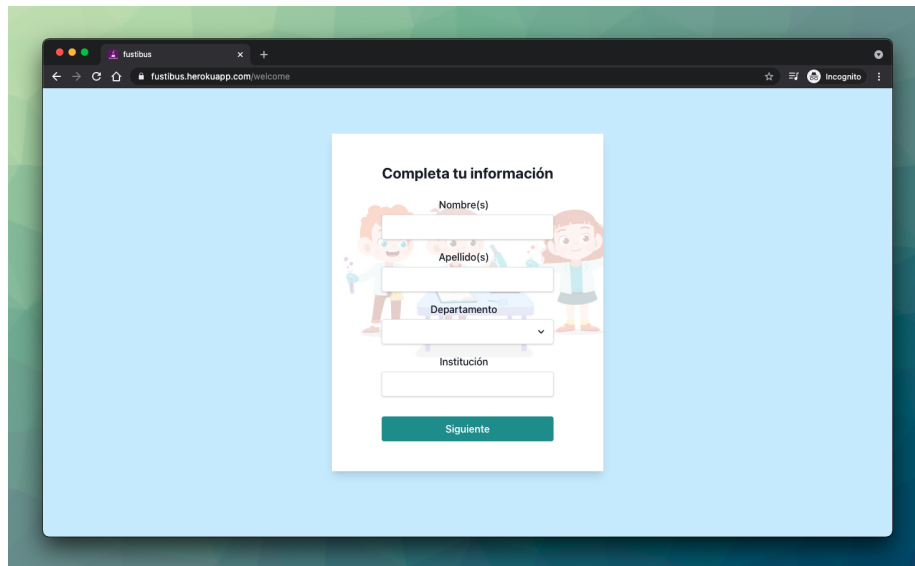


Figura 6.3: Welcome

■ Dashboard

La vista Dashboard es el componente por defecto de la aplicación. En este, se le muestra al usuario todo lo relacionado con el Club de Ciencia activo al cual pertenece. En caso de no pertenecer a ningún club, se da la opción de ir a crear un nuevo club.

A continuación, en la figura 6.4 se puede observar dicha vista.

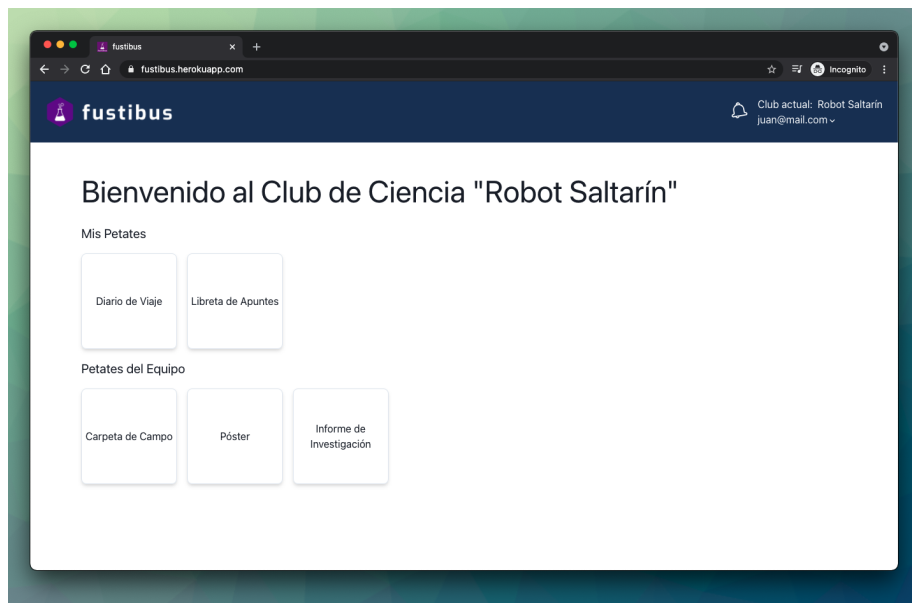


Figura 6.4: Dashboard

- Profile

En esta vista se muestra la información del usuario autenticado, permitiéndole editar la misma.

A continuación, en la figura 6.5 se puede observar dicha vista.

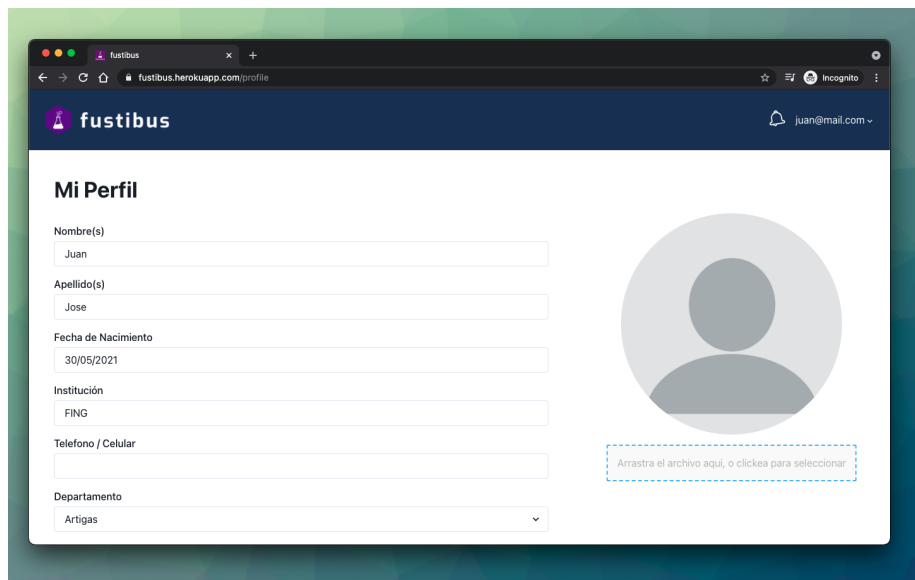


Figura 6.5: Profile

■ Clubs

En esta vista se muestra una lista de aquellos Clubes de Ciencia en los cuales el usuario actual autenticado es miembro. A partir de esta vista, se puede acceder a cada uno de dichos clubes, de forma de poder ver información más detallada de los mismos. Desde esta vista también se puede navegar para crear un nuevo club.

A continuación, en la figura 6.6 se puede observar dicha vista.

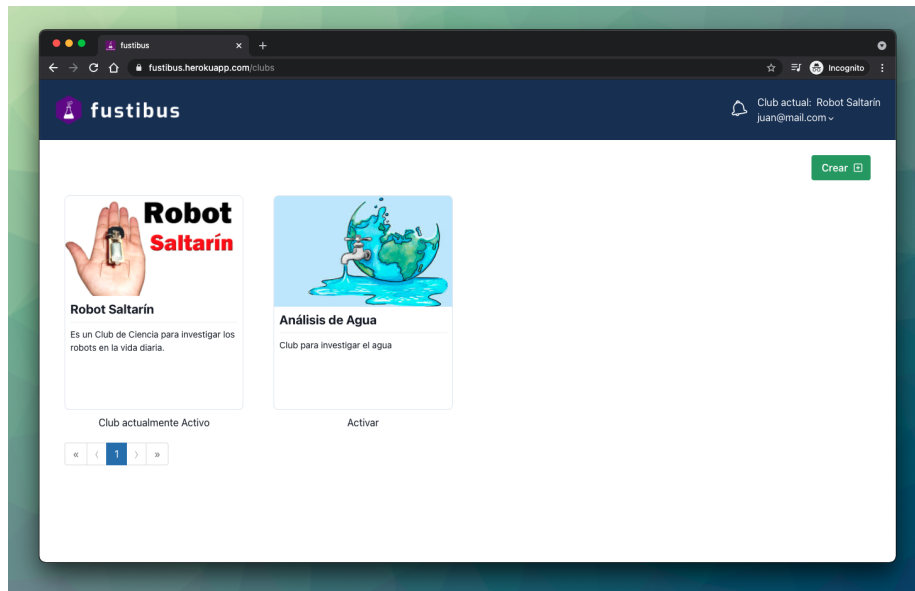


Figura 6.6: Clubs

■ Show Club

Esta vista muestra información detallada de un club de ciencia. Para poder acceder a la pantalla, el usuario debe pertenecer al club en cuestión. En caso de que el usuario tengo el rol de 'Orientador' para el club, podrá eliminar el club, así como también editarlo o invitar nuevos usuarios a formar parte.

A continuación, en la figura 6.7 se puede observar dicha vista.

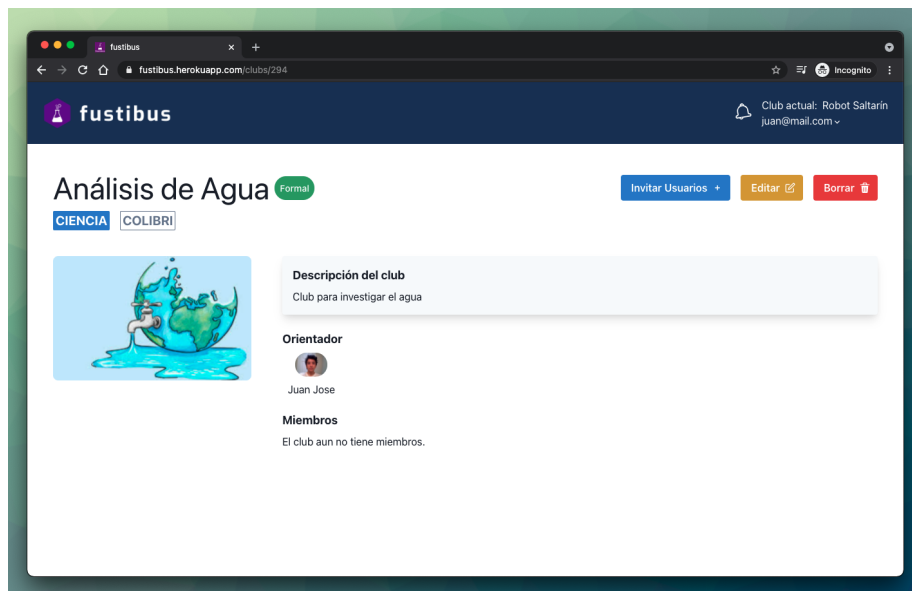


Figura 6.7: Show Club

■ Create Club

Esta vista muestra un formulario el cual permite la creación de un nuevo Club de Ciencia. Permite ingresar el nombre, la categoría y el área del club, así como también una descripción y una imagen. Es importante aclarar que, el usuario que crea dicho club será el orientador del mismo por defecto. Sin embargo, el usuario más tarde podrá hacer orientador a miembros del club si así lo desea.

A continuación, en la figura 6.8 se puede observar dicha vista.

The screenshot shows a web browser window with the URL `fustibus.herokuapp.com/clubs/new`. The page has a dark blue header with the 'fustibus' logo and a user profile 'juan@mail.com'. The main form is white and contains the following fields:

- Nombre:** A text input field.
- Categoría:** A dropdown menu with the placeholder text 'Seleccioná una categoría'.
- Área:** A dropdown menu with the placeholder text 'Seleccioná un área'.
- Descripción:** A large text area for a detailed description.
- ¿Es de educación formal?:** A toggle switch.
- Imagen:** A dashed box for uploading an image, with the text 'Arrastra el archivo aquí, o clickea para seleccionar'.

At the bottom of the form are two buttons: 'Cancelar' (red) and 'Crear' (green).

Figura 6.8: Create Club

■ Report

Esta vista permite subir un informe de forma que luego pueda ser accedido por cualquier usuario de dicho club. El usuario sube un archivo local desde su dispositivo, y se envía al Backend para que pueda ser almacenado en Cloudinary. En caso de que ya exista un informe al ingresar en esta vista, el usuario podrá descargar el archivo, abrirlo en el navegador o subir una nueva versión reemplazando la existente.

A continuación, en la figura 6.9 se puede observar dicha vista.

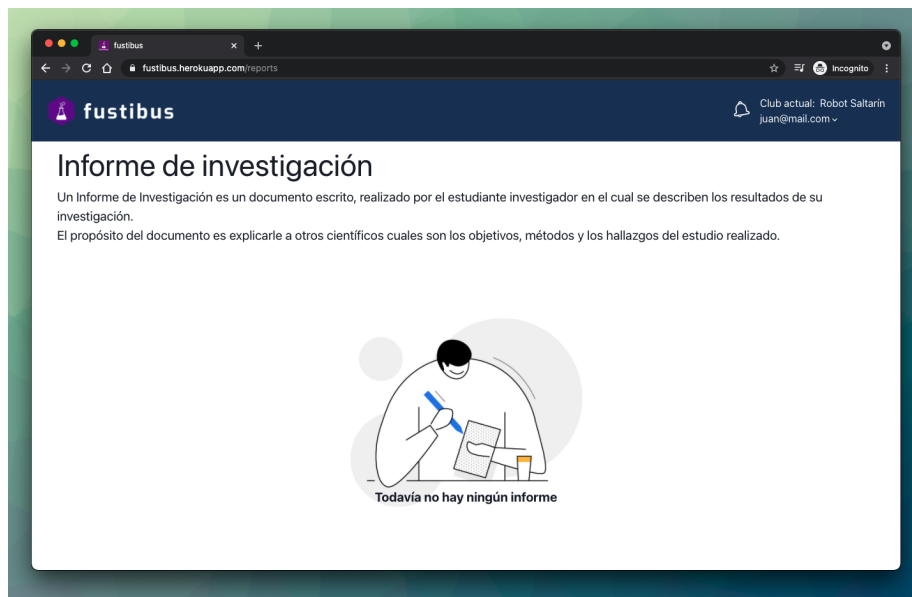


Figura 6.9: Report

■ Poster

En esta vista se muestra información y recomendaciones de como debe ser un póster científico, así como también los principales temas que debe tratar. Se trata de poder ayudar al usuario con la elaboración del mismo. También, en esta vista se puede crear un póster, guardarlo y exportarlo como PDF.

A continuación, en las figuras 6.10 y 6.11 se pueden observar las vistas del póster.

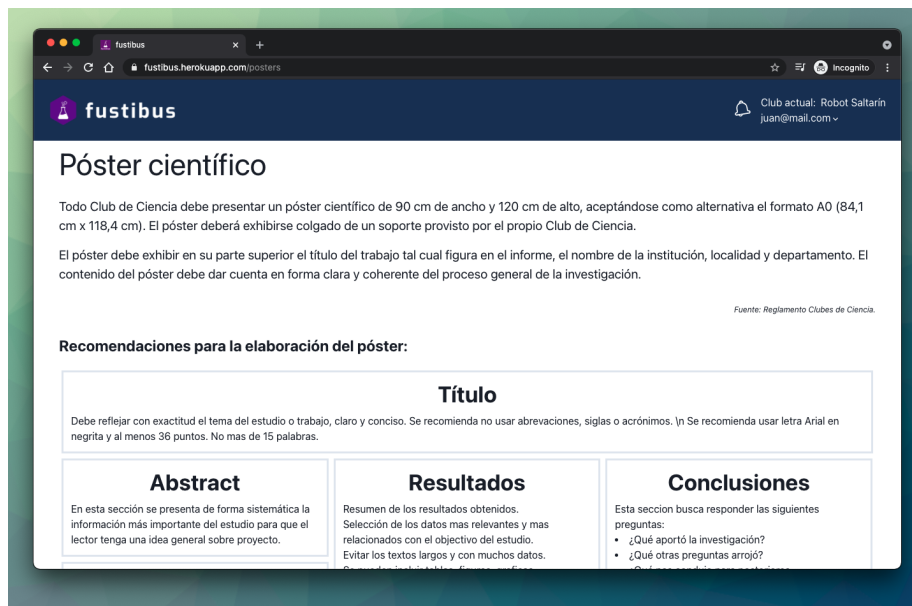


Figura 6.10: Póster - Información y recomendaciones

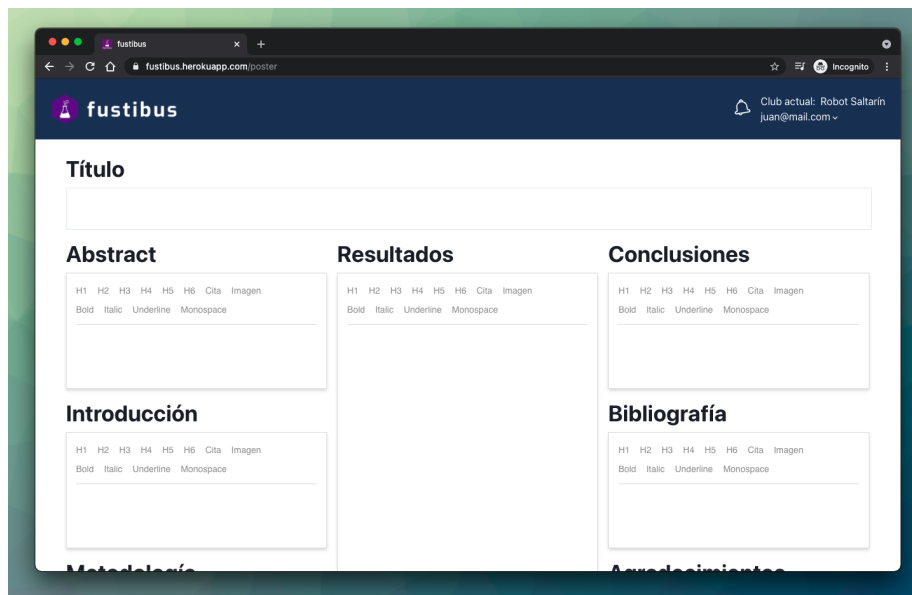


Figura 6.11: Póster - Creación

- Field Folder

Esta vista muestra todas las entradas de la Carpeta de Campo ordenadas de forma cronológica. Sin embargo, también permite filtrar por contenido y/o fecha. Desde aquí, también se puede crear una nueva entrada, así como también editar o eliminar entradas ya existentes.

A continuación, en la figura 6.12 se puede observar dicha vista.

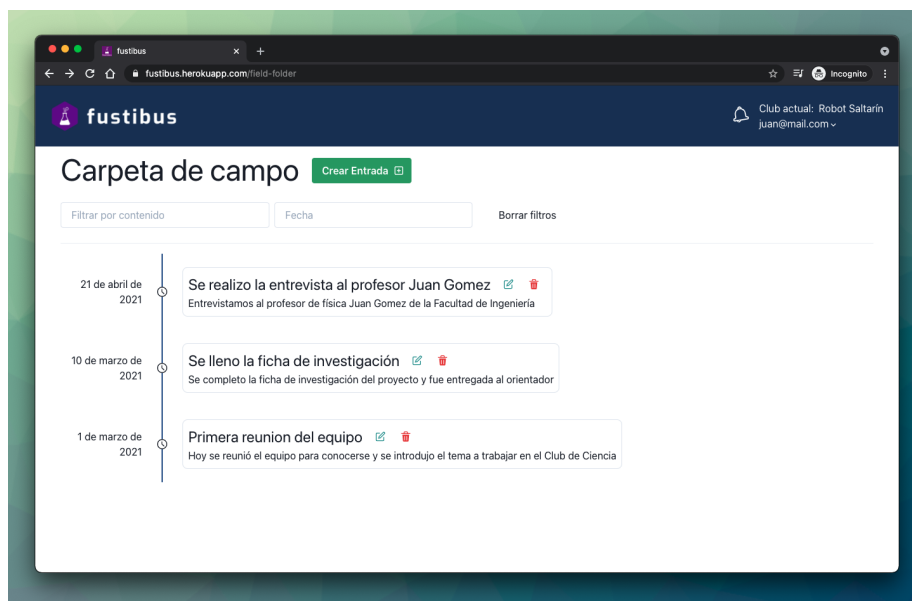


Figura 6.12: Field Folder

■ Diary

El Diario de Viaje permite organizar las tareas que tiene un club de ciencia para lograr el objetivo final. En esta vista el orientador de un club puede crear ‘misiones’, con un título y una descripción, para luego poder ir cambiándole el estado de acuerdo al estado actual de la misión. Una misión al recién crearse tiene el estado ‘pendiente’, luego puede pasar al estado ‘en progreso’, y por ultimo al estado ‘completada’. La idea general de esta vista es tener y poder visualizar una especie de *roadmap* del Club de Ciencia en cuestión. Adicionalmente, en dicha vista se puede editar y/o eliminar misiones ya existentes. Cabe destacar, que al crearse un nuevo club, se le asigna una primera misión por defecto la cual tiene como título “Primera reunión de equipo” y como descripción “El objetivo de la misión

es documentar todo lo relacionado con la primera reunión del equipo”.

A continuación, en la figura 6.13 se puede observar dicha vista.

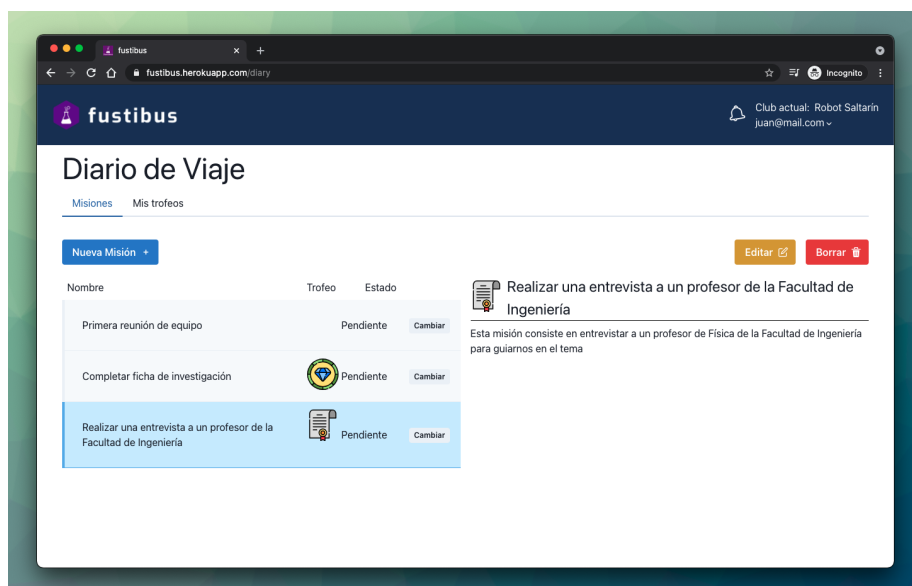


Figura 6.13: Diary

■ Notebook

La vista libreta de apuntes muestra el conjunto de notas del usuario autenticado. Para cada nota existente, permite editar o borrar cada una de ellas. Por otro lado, también se pueden crear nuevas notas.

A continuación, en la figura 6.14 se puede observar dicha vista.

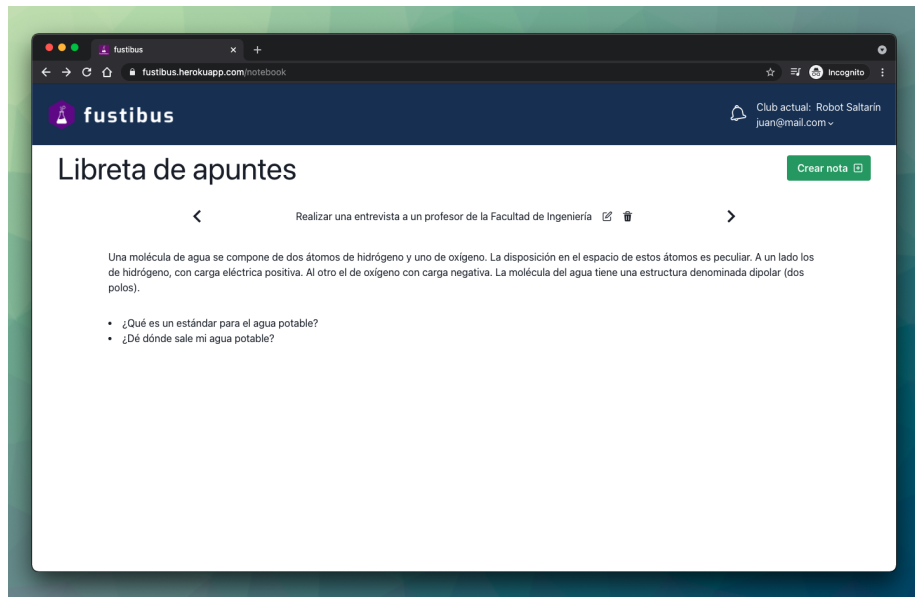


Figura 6.14: Notebook

■ New/Edit Note

En esta vista se puede crear/editar una nota. Cada nota esta asociada a una misión del Diario de Viaje, por lo que lo primero que se debe hacer al crear una nueva nota, es seleccionar para qué misión en progreso es la nota a crear. Una vez hecho esto, se procede a crear el contenido de la nota, donde se puede seleccionar agregar 'texto', 'imagen + texto', 'lista' y/o 'imagen'.

A continuación, en la figura 6.15 se puede observar dicha vista.

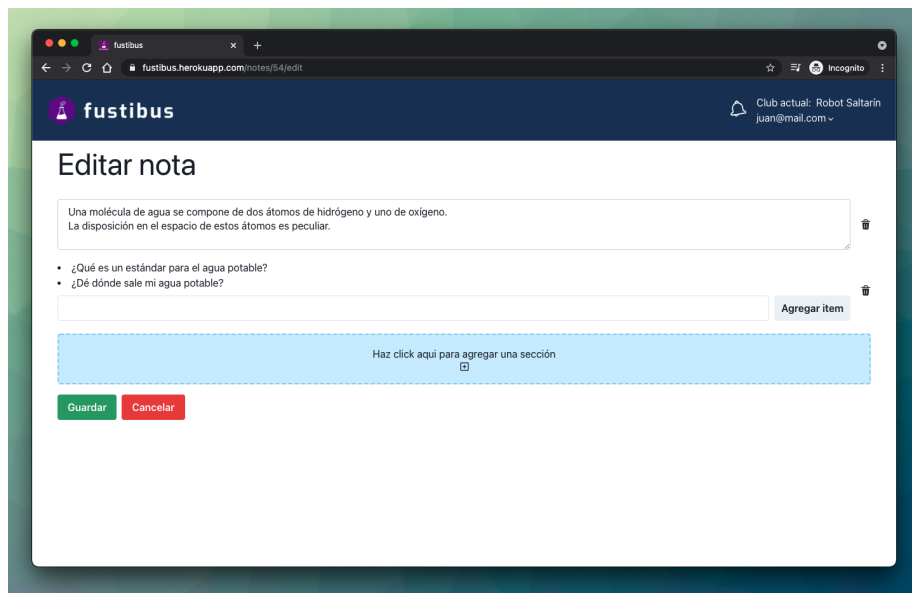


Figura 6.15: Edit Note

Anteriormente se describieron las principales vistas de la aplicación web. Sin embargo, cada una de ellas internamente utilizan componentes más pequeños los cuales son reutilizados en muchas partes de la aplicación.

El objetivo de estos componentes es el de lograr consistencia visual en todas las pantallas, y de lograr una mejor mantenibilidad de la plataforma. De tal forma de no repetir código y, en caso de querer en un futuro hacer algún cambio funcional o visual sobre un componente, alcanza solamente con cambiar el componente en cuestión, implicando de esta forma que se cambie en todas las vistas donde se utilice.

Entre los componentes más destacados se encuentran los siguientes:

- Input
- Select
- Check box
- Button
- Dropdown

- Loading
- Alert
- Modal
- Tabs
- Pagination
- Datepicker
- File Uploader

7. Despliegue de la solución

Se decidió publicar la aplicación a la nube para que esta sea accesible por nuestros tutores y también para poder realizar pruebas en un ambiente que no sea el de desarrollo. La misma se puede acceder en este enlace: <https://fustibus.herokuapp.com>.

A su vez se le compartió el enlace a Fabio Lima, un colega nuestro, el cual fue orientador de varios clubes de ciencia y su *feedback* fue fundamental para el desarrollo de este proyecto.

En este capítulo se presenta Heroku¹ en la sección 7.1, la plataforma utilizada para el despliegue de la solución en la nube, y también se realiza una comparación de Heroku con Amazon Web Services² en la sección 7.2. Por último, se explica el proceso de despliegue continuo y automatizado en la sección 7.3.

7.1. Despliegue en Heroku

La aplicación fue desplegada utilizando Heroku. Como se explica en su página web, *“Heroku ejecuta su aplicación en contenedores Linux livianos y aislados llamados ‘dynos’. La plataforma ofrece diferentes tipos de ‘dynos’ para ayudarlo a obtener los mejores resultados para su tipo de aplicación.”* [12]

Se utilizó la versión *Free* de los diferentes ‘dynos’ ofrecidos por Heroku, la misma esta enfocada a aplicaciones no comerciales, pruebas de conceptos, MVP y proyectos personales, por lo cual esta versión resultó más que adecuada para lo que se estaba buscando resolver.

¹<https://www.heroku.com>

²<https://aws.amazon.com>

7.2. Comparación con otras plataformas [28] [19]

Al decidir la plataforma sobre la que se alojaría la aplicación se consideraron dos alternativas, estas fueron Amazon Web Services (AWS) y Heroku.

Por un lado, AWS es una plataforma *Infrastructure as a Service* (IaaS), la cual proporciona docenas de servicios que ayudan a los usuarios a lidiar con bases de datos, administración, análisis, implementación y otras tareas para el despliegue de sus aplicaciones. Se puede decir entonces que la plataforma AWS es una combinación de varias infraestructuras de computación en la nube que ofrecen productos y servicios.

Por otro lado, como ya se explicó anteriormente, Heroku es una plataforma, *Platform as a Service* (PaaS), en la nube basada en contenedores la cual permite fácilmente desplegar y escalar aplicaciones web.

Este modelo de computación en la nube cuenta con una web para la creación de aplicaciones fácil y rápida. Se le libera al desarrollador de las tareas relacionadas con los servidores, la virtualización, el almacenamiento y las redes; el proveedor es responsable de ellas. Por lo tanto, el desarrollador de la aplicación no tiene preocupaciones relacionadas con los sistemas operativos, middleware, actualizaciones de software, etc. Los recursos digitales, como CPU, RAM y otros, se pueden administrar a través de un panel administrativo visual.

En general, este servicio de tecnología en la nube es más fácil de usar en comparación con los servicios ofrecidos por AWS que se han mencionado.

La principal diferencia entre las plataformas AWS y Heroku radica en que la primera de ellas es un servicio IaaS, mientras que la segunda brinda un servicio PaaS. Una plataforma IaaS implica la configuración manual del servidor y la aplicación, mientras que PaaS proporciona herramientas listas para usar pero no permite administrarlas.

Para el caso de nuestro proyecto, la mejor es la solución PaaS ya que simplemente se debe elegir la configuración requerida por nuestras necesidades sin preocuparnos por cosas que nuestro proyecto aún no necesita, y en caso de necesitarlas en el futuro, se puede evaluar actualizar las características del plan a uno mejor. Es por ello que se decidió utilizar Heroku para el despliegue de nuestra aplicación.

7.3. Despliegue continuo y automatizado [\[18\]](#)

Heroku se integra con GitHub para facilitar la implementación del código en aplicaciones que se ejecutan en la plataforma. Cuando la integración de GitHub está configurada para una aplicación, Heroku puede compilar y desplegar automáticamente la aplicación.

Debido a que nuestro código está versionado en GitHub, se configuró para que cada vez que se haga una actualización sobre la *branch* “master”, la aplicación automáticamente fuera desplegada siempre y cuando los test de integración continua hayan sido satisfactorios.

8. Calidad del software

En esta sección se describen los diferentes mecanismos que se llevaron a cabo para asegurar la calidad del software implementado.

En la sección 8.1 se explica el proceso seguido para la revisión de código, luego, en la sección 8.2 se presentan las pruebas unitarias realizadas para asegurar la correctitud del software implementado. Después, en la sección 8.3 se presentan las pruebas de integración que se realizaron, y por último, en la sección 8.4 se detalla sobre la validación con usuarios.

8.1. Revisión de código

Al momento de realizar una nueva funcionalidad, el integrante a cargo de realizarla crea una nueva *branch* en su ambiente de desarrollo. Posteriormente, una vez finalizado el trabajo, el integrante crea un *pull-request*¹ en la plataforma GitHub, donde está alojado el código.

El *pull-request* queda disponible para ser revisado por cualquier otro integrante del equipo. Si bien habían cambios pequeños los cuales no era necesario revisar, en general, los cambios no podían ser integrados si no tenían al menos una aprobación por parte de un miembro del equipo.

Este proceso de revisión de código permite corroborar los detalles que no son cubiertos por las pruebas unitarias, como puede ser la repetición de código, código innecesario, malas prácticas de programación y malas decisiones de

¹En su forma más simple, los *pull-request* son un mecanismo para que un desarrollador notifique a los miembros del equipo que ha completado una funcionalidad. Una vez que está lista, el desarrollador crea un *pull-request*. Esto les permite a todos los involucrados saber que necesitan revisar el código. Si hay algún problema con los cambios, los compañeros de equipo pueden publicar comentarios. [29]

desarrollo, evitando así que sean integradas a la *branch* principal, “master”.

En la imagen 8.1 se puede ver un ejemplo de revisión de código.

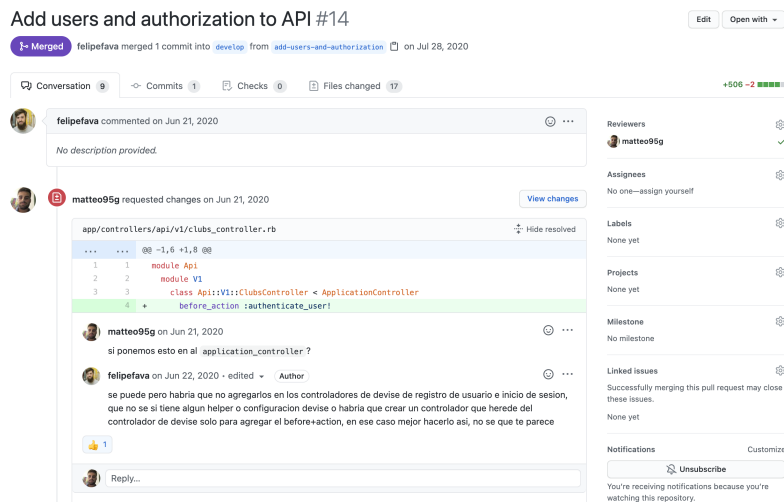


Figura 8.1: Ejemplo de revisión de código

8.2. Pruebas unitarias

Como se explica en la web de Microsoft, *se llaman pruebas unitarias porque descompone la funcionalidad de su programa en comportamientos discretos comprobables, que se pueden probar como unidades individuales.* [24]

Tan pronto como un integrante del equipo agrega una nueva funcionalidad o bloque de código en la aplicación, corría por su cuenta crear pruebas unitarias. Se crearon pruebas unitarias para verificar el comportamiento del código, utilizando datos de entrada correctos e incorrectos y corroborando cualquier suposición explícita o implícita hecha por el código, comparando la salida real del bloque de código a probar contra la salida esperada.

Se decidió hacer pruebas unitarias solamente para el backend, ya que se consideró que los componentes visuales podrían ser probados mediante *QA (Quality Assurance)* o pruebas de integración, que se explican en el próximo punto. Para realizar las pruebas unitarias se utilizó RSpec, un framework de testing basado en comportamiento para Ruby.

Los resultados de las pruebas unitarias se pueden ver en el anexo E.

8.3. Pruebas de integración *end-to-end*

Asegurarse de que todos los módulos y métodos implementados funcionen juntos correctamente es el objetivo de las pruebas de integración. Las pruebas *end-to-end* (E2E) son un tipo de pruebas de integración, e involucran técnicas que simulan un usuario real de una aplicación. Al replicar las acciones que realizaría un usuario, la prueba ayuda a evaluar si los resultados cumplen con los requisitos o el resultado esperado.

Estas pruebas se ejecutan en un entorno que idealmente representa el entorno de producción en términos de bases de datos utilizadas, servicios y tipo de dispositivo utilizado. Las pruebas E2E implican probar el flujo de la aplicación y ayudan a descubrir una falla en el flujo. Una de estas fallas determinaría un error en la aplicación. [2]

Hay dos formas de llevar a cabo este tipo de pruebas, una de ellas es de forma manual y la otra es de forma automatizada.

Las pruebas manuales fueron realizadas por cada miembro del equipo luego de realizar una nueva funcionalidad o cambio en el código. El encargado de realizar la prueba seguiría el mismo proceso que haría un usuario del sistema al utilizar la aplicación, de forma de validar que todo funcione de manera esperada.

Se realizaron también pruebas automatizadas E2E, para ello se utilizó la herramienta Ghost Inspector². Ghost Inspector es una herramienta automatizada de monitoreo y pruebas E2E. Permite grabarse a sí mismo realizando acciones en el sitio web a probar dentro del navegador, luego sincronizarlas con el servicio Ghost Inspector y ejecutarlas.

Las pruebas pueden ser ejecutadas automáticamente a través de la API que ofrece Ghost Inspector cuando hay cambios en el código, o pueden ser grabadas utilizando el sitio y ser ejecutadas automáticamente en un intervalo establecido para verificar continuamente si hay problemas. Para el caso de nuestro proyecto, las pruebas fueron creadas utilizando la extensión que ofrece Ghost Inspector para el navegador Google Chrome. La extensión permite grabar las acciones ejecutadas desde el navegador y luego realizar aserciones para verificar el correcto funcionamiento de la aplicación.

La herramienta permite también ver la grabación de los pruebas que fueron ejecutadas y las acciones que se llevaron a cabo de forma automática en la

²www.ghostinspector.com

aplicación web que ofrece. Se pueden ver los resultados de las pruebas ejecutadas utilizando Ghost Inspector en el anexo [F](#).

8.4. Validación con usuarios

De manera de validar nuestro trabajo, se le solicitó a Fabio Lima, como mencionamos anteriormente, que pruebe la herramienta y nos brinde comentarios. Él a su vez lo compartió dentro de su círculo de trabajo al cual pertenecen otros colaboradores orientadores de Clubes de Ciencia con el fin de abarcar una mayor cantidad de usuarios.

Se obtuvo como devolución de carácter general el siguiente comentario: *“La aplicación tiene potencial no solo para los Clubes de Ciencia sino que para el trabajo en proyectos en general. Es una herramienta útil para compartir notas, apuntes, fotos, etc. Como versión inicial está muy bien lograda, aunque tiene cosas a mejorar...”*

Esto nos confirmó que el camino elegido fue correcto, que brinda una buena solución a la necesidad encontrada y que la herramienta tiene lugar para ser mejorada.

Con respecto a las mejoras, surgieron algunas consultas o sugerencias, así como también algunos *bugs* a corregir. A continuación, se lista una breve descripción u observación al respecto de los mismos:

- *Consulta si al momento del registro, el usuario queda asociado a una única institución.*

Esta consulta surge porque por lo general, los docentes trabajan en varias instituciones y se sugirió que exista un perfil de usuario que pueda asociarse a más de una institución o grupo.

Desde nuestro lado, el campo institución en realidad se vio solo como un campo *String* asociado al usuario, el cual no tiene una asociación con los clubes. Es decir, el propósito es solo informativo y no interfiere con la participación de un usuario un uno o varios clubes.

De todas maneras es algo que se podría considerar para mejorar o al menos aclarar, de forma tal de no generar dudas a los usuarios de la aplicación.

- *Sugerencia para que las categorías de los clubes se ordenen por edad en vez de alfabéticamente y aclarando las edades en el listado.*

Esto es debido a que las categorías en realidad representan franjas etarias las cuales compiten y son evaluadas entre si.

Es una observación totalmente válida la cuál se debería tomar en cuenta como mejora futura.

- *Duda acerca del campo “¿Educación formal?”, el cual no estaba claro a qué se refería.*

Este campo está presente en el reglamento de las ferias de Clubes de Ciencias, el cual a su vez está relacionado a una categoría que solo está habilitada para este tipo de educación y refiere a clubes que se creen dentro de una institución educativa, y no así a los formados por individuos por fuera de ellas.

Como mejora a dicho comentario, se debería agregar información detallando el propósito de dicho campo en el formulario de creación de un club.

- *Problema de navegación al finalizar la creación de un club.*

Esto estaba detectado como *bug* en nuestro tablero de trabajo y fue corregido luego de recibir el comentario.

- *Sugerencia para que al crear un usuario se permita elegir si se es estudiante o docente y así tener distintos permisos.*

Fue una decisión de diseño que los usuarios al crearse sean todos usuarios “generales” o “estudiantes”, y que al crear un club, el usuario creador se convierta en “orientador” o “docente” de dicho club.

Dicho usuario luego tiene el permiso para convertir a otros usuarios del club a orientadores.

- *Duda acerca de como volver a ver la página principal.*

Esta página principal solo se ve cuando no sos participante de ningún club. Una vez que un usuario crea uno o es invitado a uno ya el “Home” de la aplicación pasa a ser la vista del club que se tiene como “activo”.

- *Duda sobre los trofeos, ¿deberían ser entregados por los “orientadores”?*

La idea inicial de los trofeos era que cada misión tuviera uno asociado y que los participantes que la completaran, ganaran el trofeo de manera de tener una competencia sana por completar las misiones.

Sin embargo, lo finalmente desarrollado fue que cuando el orientador marca la misión como completa, el trofeo es ganado por el club, de forma grupal y no individual.

9. Gestión

Este capítulo describe la gestión del proyecto, se describe la metodología de trabajo del equipo en la sección 9.1, en la sección 9.2 se describe la comunicación del equipo y luego en la sección 9.3 se presentan las herramientas utilizadas.

9.1. Metodología de trabajo

El desarrollo del sistema fue realizado de forma iterativa e incremental, se adoptó un esquema de desarrollo ágil.

9.1.1. Metodología ágil

Por definición, *“el desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos auto-organizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo.”*¹

Las cuatro características principales que son fundamentales para todas las metodologías ágiles son: planificación adaptativa, desarrollo iterativo y evolutivo, respuesta rápida y flexible al cambio, y promover la comunicación. [27]

En esencia, agilidad significa responder a los cambios rápida y eficientemente. Los posibles cambios requeridos en los proyectos de software se encuentran en el presupuesto, el cronograma, los recursos, la tecnología, los requisitos y el equipo. Estos son los cambios “reactivos” en los que se centran los modelos

¹https://es.wikipedia.org/wiki/Desarrollo_agil_de_software

ágiles entregando el primer incremento en las primeras semanas y el software completo en las semanas o meses siguientes. [32]

Beneficios de la metodología ágil [27]

- Manejo del cambio de requisitos: La fase de planificación se mejora drásticamente. Primero, debido a que los clientes están directamente involucrados en el proceso de desarrollo, es decir, los clientes controlan los procesos de los proyectos, los requisitos reflejan realmente las necesidades actuales de los usuarios finales.
- Detección de fallas: A medida que se realizan pruebas durante cada iteración, las fallas se detectan antes y se pueden corregir previo a que aumenten en gravedad. Además, las pruebas continuas permiten una retroalimentación permanente, lo que mejora aún más el código desarrollado en iteraciones futuras.
- Mayor rendimiento: Las reuniones diarias brindan la oportunidad de intercambiar información valiosa y ajustar las mejoras continuamente. La capacidad de discutir y compartir ideas fomenta el trabajo en equipo. Una mejor comunicación conduce a un mayor intercambio de conocimientos, equipos autoorganizados y aumento en la moral del equipo a medida que los empleados comienzan a confiar y a ganarse la confianza de sus compañeros de trabajo. Esto aumenta la productividad y genera un mejor rendimiento en términos de retorno de la inversión que la suma de toda la producción individual.
- Entrega iterativa e incremental: La entrega del proyecto se divide en pequeñas versiones funcionales o incrementos para gestionar el riesgo y obtener comentarios tempranos de los clientes y usuarios finales. Estas pequeñas versiones se entregan según un cronograma mediante iteraciones que suelen durar entre una y cuatro semanas cada una. El progreso de la funcionalidad del software se puede verificar y monitorear con mucha más frecuencia que al final de los hitos largos.
- Flexibilidad de diseño: El diseño debe ser flexible para que pueda adaptarse a los cambios fácilmente.

- Mejora de la calidad: Se utiliza el desarrollo basado en pruebas y la refactorización. La refactorización conduce a una mayor reutilización del código y a una mejor calidad. Se mejoran todos los aspectos del software, desde el diseño y la arquitectura hasta el rendimiento de los productos de cada sprint.

9.1.2. Kanban

Específicamente, se adoptó la metodología Kanban dentro de las metodologías ágiles. En la pagina web de Atlassian Agile Coach² se define a Kanban como *“un framework popular utilizado para implementar en el desarrollo de software ágil y DevOps. Requiere comunicación en tiempo real y total transparencia del trabajo. Los elementos de trabajo se representan visualmente en un tablero Kanban, lo que permite a los miembros del equipo ver el estado de cada trabajo en cualquier momento.”*

Tablero Kanban [25]

Las tareas de todos los equipos que siguen esta metodología giran en torno a un tablero Kanban, una herramienta que se utiliza para visualizar el trabajo y optimizar el flujo del mismo entre el equipo. Si bien los tableros físicos son populares entre algunos equipos, los tableros virtuales son una característica crucial en cualquier herramienta de desarrollo de software ágil por su trazabilidad, fácil colaboración y accesibilidad desde múltiples ubicaciones.

Independientemente de si el tablero es físico o digital, su función es garantizar que se visualice el trabajo del equipo, que el flujo de las tareas esté estandarizado y que todas las bloqueantes y dependencias se identifiquen y resuelvan de inmediato. Un tablero Kanban básico tiene un flujo de tres pasos: Pendiente, En curso y Hecho. Sin embargo, se pueden agregar más pasos en el proceso, dependiendo de las características del proyecto.

Kanban se basa en la transparencia total del trabajo y la comunicación de la capacidad de un equipo en tiempo real, por lo tanto, el tablero debe verse como la única fuente de verdad para el estado de las tareas.

En la figura 9.1, se ve una captura de pantalla del tablero utilizado para el proceso de desarrollo. La herramienta utilizada fue *Trello*, explicada en la

²<https://www.atlassian.com/agile/kanban>

sección 9.3.

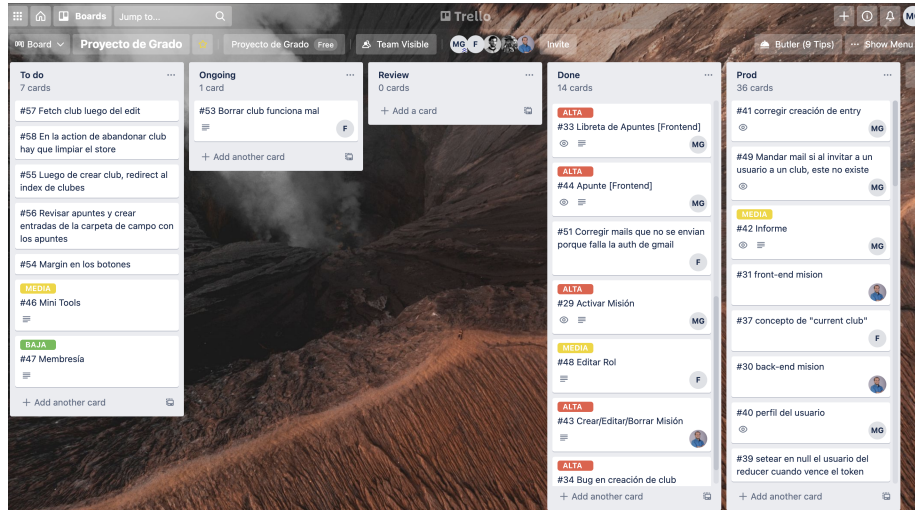


Figura 9.1: Tablero Kanban

En japonés, Kanban se traduce literalmente como “señal visual”. Para los equipos Kanban, cada elemento de trabajo se representa como una tarjeta separada en el tablero.

El principal propósito de representar el trabajo como una tarjeta en el tablero es permitir que los miembros del equipo rastreen el progreso a través de su flujo de una manera visual.

Las tarjetas presentan información crítica sobre ese elemento en particular, lo que brinda a todo el equipo una visibilidad completa de quién es responsable del mismo, una breve descripción de lo que se está realizando, cuánto tiempo se estima que tomará y demás datos relevantes.

En nuestro caso en concreto, nuestro tablero se conformó por las siguientes etapas o columnas:

- To Do

Conteniendo las tareas a realizar, ordenadas por prioridad. Es decir, las tareas de más arriba de dicha columna son más prioritarias que las de más abajo.

- Ongoing

Las tareas en esta columna son aquellas que están en progreso. Desde esta columna en adelante, es muy importante que todas las tareas tengan un miembro del equipo asignado.

- Review

En esta columna, deben estar las tareas que ya fueron desarrolladas y están listas para ser revisadas por algún miembro del equipo.

- Done

Aquí están todas las tareas que ya fueron revisadas y aprobadas, y el código ya fue introducido en la rama estable del proyecto.

- Production

Por último, esta columna representa aquellas tareas que ya fueron desplegadas en el ambiente de producción en la nube.

9.2. Comunicación

En cuanto a la comunicación del equipo, en las primeras etapas del proyecto se realizaron reuniones presenciales para comenzar el trabajo. En estas reuniones se fueron definiendo los pasos a seguir y también tuvieron lugar algunas de las primeras tareas del descubrimiento del producto.

Con el proyecto un poco más avanzado, se nos presentó la dificultad de cambiar este tipo de instancias presenciales a instancias remotas, debido a la pandemia ocasionada por el virus SARS-CoV-2 causante de la enfermedad COVID-19. Esto llevó a organizarnos en reuniones virtuales, las que afortunadamente son algo cotidiano en nuestro día a día debido a nuestros trabajos en desarrollo de software con clientes en el exterior.

En estas reuniones fueron participando tutor y co-tutor cuando era necesario y en las restantes solo el equipo del proyecto. En varias estuvo presente también Gonzalo Vilar quién nos guió durante el proceso del *Design Sprint* remoto.

La comunicación escrita se dio a través de e-mails y de la herramienta *Slack*, siendo la primera la principal herramienta para comunicación formal y de organización general, y la segunda de carácter informal para el manejo diario del proyecto entre nosotros.

En la próxima sección se explica en más detalle qué es *Slack*, junto con las demás herramientas utilizadas para la gestión del proyecto.

9.3. Herramientas

Para llevar a cabo el desarrollo del sistema se utilizaron herramientas para organizar el trabajo, visualizar tareas pendientes y avance de la implementación. Entre las herramientas utilizadas se encuentran Trello, Git, GitHub, Slack, Google Drive y Google Hangouts.

9.3.1. Trello

Trello³ es una herramienta colaborativa, que permite organizar proyectos en tableros, creando listas de estilo Kanban. De esta forma, con Trello se puede ver en qué se está trabajando, quién está trabajando en qué y cuales son las tareas en proceso. Trello es una subsidiaria de Atlassian⁴. [34]

9.3.2. Git

Para el versionado del código se utilizó Git⁵. Git es un software libre y de código abierto, creado para llevar el control de versiones desde proyectos pequeños a muy grandes, con velocidad y eficiencia. [17]

9.3.3. GitHub

GitHub⁶ es una plataforma que permite alojar código gratuitamente, para el control de versiones y colaboración del equipo de desarrollo. Permite a los desarrolladores trabajar en proyectos desde cualquier parte del mundo. [20]

9.3.4. Slack

Slack⁷ es una plataforma de comunicación empresarial, desarrollada por la empresa de software estadounidense Slack Technologies. Slack ofrece muchas

³<https://trello.com/>

⁴<https://www.atlassian.com/>

⁵<https://git-scm.com/>

⁶<https://github.com/>

⁷<https://slack.com/>

funciones como salas de chat (canales), grupos privados y mensajería directa. [40]

9.3.5. Google Drive

Google Drive⁸ es un servicio de sincronización y almacenamiento de archivos desarrollado por Google. Permite a los usuarios almacenar archivos, sincronizar archivos entre dispositivos y compartir archivos. [35]

9.3.6. Google Hangouts

Google Hangouts⁹ es una aplicación de mensajería y llamadas de voz multiplataforma desarrollada por Google. Hangouts, que originalmente era una función de Google+, se convirtió en un producto independiente en 2013. [36]

⁸<https://drive.google.com/>

⁹<https://hangouts.google.com/>

10. Conclusiones

En este capítulo se analizan los resultados obtenidos a partir del trabajo realizado, y se presentan posibles opciones de trabajo a futuro que permitirán darle continuidad al proyecto.

10.1. Resultados y discusión

La ejecución de este proyecto deja como resultado una plataforma de trabajo colaborativo orientada a los Clubes de Ciencia. La misma es una aplicación web la cual permite la creación de un club, y la gestión del mismo. Cada club, está compuesto por uno o más orientadores, y los integrantes de dicho club.

Adicionalmente, la utilización de esta plataforma permite una alta disponibilidad de los recursos relacionados a los Clubes de Ciencias por parte de los integrantes. Busca centralizar la información en un sólo lugar, y permitir a los usuarios colaborar para llevar a cabo el objetivo del club al cual pertenecen.

Otro aspecto que se cree muy importante resaltar es que la plataforma puede ser usada como herramienta de colaboración remota. Un dato no menor dada la situación sanitaria mundial de pandemia que estamos viviendo actualmente.

Respecto al uso y a las funcionalidades de la plataforma, ésta permite a un usuario crear un Club de Ciencia, e invitar a usuarios a formar parte del mismo. El usuario creador del club, tendrá el rol de orientador dentro de dicho club. Una vez creado, se deberán de crear misiones. Las misiones son actividades o consignas que tienen una descripción y son ingresadas por el orientador. Dichas misiones pueden tener asignado un trofeo, el cual se obtendrá como recompensa al completar dicha misión. A su vez, para las misiones pendientes se espera que los integrantes del club produzcan un resultado escrito, a modo de apunte, además del resultado tangible. Esto se logra mediante la libreta de apuntes.

Por otro lado, la plataforma permite tener asociado al club la Carpeta de Campo. Éste es un documento donde el Club de Ciencia registra cada paso que da en la investigación en orden cronológico, y es el único indicador que da muestra del proceso de forma oficial.

Adicionalmente, la herramienta da la posibilidad de guardar el Informe de Investigación asociado al club, de forma de tenerlo en un lugar compartido y de fácil acceso para cualquier integrante. Dicho informe es el instrumento escrito a través del cual el Club de Ciencia comunica los resultados de la investigación.

Finalmente, la plataforma permite la creación de un póster científico, el cual cada club deberá presentar en la exhibición de los proyectos.

10.2. Trabajo futuro

Es importante poder identificar el trabajo realizado y los resultados obtenidos, pero también identificar posible trabajo a futuro para dar continuidad al esfuerzo invertido. Esta sección pretende identificar el trabajo futuro necesario para seguir ampliando las funcionalidades y el alcance de la aplicación implementada.

Los siguientes puntos corresponden a mejoras o nuevas funcionalidades las cuales se considera que pueden ser llevadas adelante como trabajo a futuro.

- Mejorar la interfaz gráfica, de forma que la usabilidad y la accesibilidad de la misma se vea incrementada. Lo anteriormente mencionado podría ser resuelto solicitando asesoramiento de un especialista en diseño de interfaces gráficas con experiencia en el área.
- Ampliar la funcionalidad de invitación a un Club de Ciencia, permitiendo que un miembro pueda visualizar los clubes disponibles y enviar una solicitud para unirse a un club.
- Integración con el MEC al momento de crear un Club de Ciencia, de forma de que al crear el mismo también quede inscripto como se indica en el reglamento. Esto se podría resolver enviando un e-mail automatizado con los requerimientos solicitados por el MEC para la inscripción de dicho club.
- Muchas de las vistas de la aplicación son compatibles con pantallas de

dispositivos móviles, sin embargo, muchas otras no. Queda pendiente optimizar las vistas en las que su disposición en pantallas móviles no es la adecuada. Se propone revisar las diferentes vistas y optimizarlas para dispositivos móviles.

- En caso de que el funcionamiento de la aplicación web se vea deprecado debido a que no esta optimizada para ser ejecutada en un dispositivo móvil, se puede considerar implementar una aplicación móvil.
- Presentar la plataforma a la ANEP y al MEC, buscando que sea adoptada como la herramienta por defecto de los Clubes de Ciencia de las escuelas públicas de Uruguay.
- Ampliar el alcance de la herramienta a otros países, de forma que la plataforma este disponible a Clubes de Ciencia mas allá de Uruguay.
- La implementación o integración con un chat, para que los integrantes puedan interactuar en tiempo real compartiendo ideas dentro de la plataforma. Para el caso de querer implementarlo, sugerimos que sea mediante Websockets. Por otro lado, en caso de querer integrarse con un chat sugerimos utilizar Twilio¹.
- Un editor de texto en tiempo real WYSIWYG², similar a lo que ofrece Google Docs³ para el informe de investigación y también para el póster científico. Se podría utilizar la herramienta Swellrt⁴ para realizar lo mencionado.
- Aumentar la cantidad de información que brinda la plataforma en relación a los Clubes de Ciencia y el proceso de participación de las distintas ferias. Por ejemplo, agregar en forma de *tips* datos sobre como crear el soporte para el póster, la disposición de un *stand*, etc. Como trabajo a futuro, se podrían crear nuevas secciones en la aplicación de carácter informativo.

¹<https://www.twilio.com/>

²WYSIWYG, acrónimo de What You See Is What You Get (en español, "lo que ves es lo que obtienes"), es una frase aplicada a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento mostrando directamente el resultado final. <https://es.wikipedia.org/wiki/WYSIWYG>

³<https://www.google.com/docs/about/>

⁴<https://swellrt.org/>

Referencias

- [1] *33 Feria Nacional de Clubes de Ciencia*. URL: <https://www.gub.uy/ministerio-educacion-cultura/sites/ministerio-educacion-cultura/files/documentos/publicaciones/33-feria-clubes-de-ciencia.pdf>. (accedido en: 08.05.2021).
- [2] *A comprehensive Guide to End to End (E2E) Testing*. URL: <https://www.perfecto.io/blog/comprehensive-guide-end-end-e2e-testing>. (accedido en: 02.03.2021).
- [3] *Acerca de Ruby*. URL: <https://www.ruby-lang.org/es/about/>. (accedido en: 17.02.2021).
- [4] *Affinity Diagram – Kawakita Jiro or KJ Method*. URL: <https://project-management.com/affinity-diagram-kawakita-jiro-or-kj-method/>. (accedido en: 13.03.2021).
- [5] Sanchit Aggarwal. “Modern Web-Development using ReactJS.” En: *International Journal of Recent Research Aspects* 5.1 (2018), págs. 133-137. ISSN: 23497688. URL: <http://proxy.timbo.org.uy/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edb&AN=129311347&lang=es&site=eds-live>.
- [6] *Arquitectura de tres niveles*. URL: https://www.ecured.cu/Arquitectura_de_tres_niveles. (accedido en: 20.03.2021).
- [7] Albert Campillo. *¿Qué es React y para qué sirve?* URL: <https://www.drauta.com/que-es-react-y-para-que-sirve>. (accedido en: 14.02.2021).
- [8] Guillermo Canabal, Guzmán Oholeguy y Camila Rosso. “Microscopio Mágico”. En: (2019). DOI: [20.500.12008/23214](https://doi.org/10.500.12008/23214). Proyecto de grado de Ingeniería en Computación. Universidad de la República, Uruguay.

- [9] Humberto Cervantes. *Arquitectura de Software*. Software Guru - vol. 27. URL: <https://sg.com.mx/revista/27/arquitectura-software>. (accedido en: 20.03.2021).
- [10] Mag. Rosana Álvarez Chappore. “Educación y tecnología en el Uruguay - Una mirada desde la investigación”. En: (2015).
- [11] *Cloudinary - About Us*. URL: <https://www.cloudinary.com/about>. (accedido en: 19.02.2021).
- [12] *Dyno Types*. URL: <https://www.devcenter.heroku.com/articles/dyno-types>. (accedido en: 09.02.2021).
- [13] Dirección General de Educación Inicial y Primaria. *Congreso de Clubes de Ciencia de Cultura Científica del departamento de Montevideo*. URL: <https://www.dgeip.edu.uy/1776/>. (accedido en: 12.08.2020).
- [14] Ministerio de Educación y Cultura. *Club de Ciencia*. URL: <https://www.gub.uy/ministerio-educacion-cultura/politicas-y-gestion/clubes-ciencia>. (accedido en: 12.08.2020).
- [15] Ministerio de Educación y Cultura. *Reglamento de ferias estandarizadas*. URL: <https://www.gub.uy/ministerio-educacion-cultura/comunicacion/publicaciones/reglamento-0>. (accedido en: 11.01.2021).
- [16] *Entrevista a Robert Álvarez*. URL: <https://drive.google.com/file/d/141JyEFnAo34WJtePcJi9eoBw8xeyv5kT/view?usp=sharing>.
- [17] *Git -distributed-even-if-your-workflow-isnt*. URL: <https://git-scm.com/>. (accedido en: 17.03.2021).
- [18] *GitHub Integration (Heroku GitHub Deploys)*. URL: <https://www.devcenter.heroku.com/articles/github-integration>. (accedido en: 11.02.2021).
- [19] Artur Hebda. *Heroku vs. AWS - Which PaaS Hosting to Choose?* URL: <https://www.railsware.com/blog/heroku-vs-aws-which-paas-hosting-to-choose>. (accedido en: 11.02.2021).
- [20] *Hello World*. URL: <https://guides.github.com/activities/hello-world/>. (accedido en: 17.03.2021).
- [21] *Heroku - What*. URL: <https://www.heroku.com/what>. (accedido en: 11.02.2021).
- [22] *https://About CircleCI*. URL: <https://www.circleci.com/docs/2.0/about-circleci/>. (accedido en: 19.02.2021).

- [23] *Imagine what you could build if you learned Ruby on Rails...* URL: <https://www.rubyonrails.org>. (accedido en: 17.02.2021).
- [24] Mike Jones, Gordon Hogenson y et al. *Unit test basics*. URL: <https://www.docs.microsoft.com/en-us/visualstudio/test/unit-test-basics?view=vs-2019>. (accedido en: 28.02.2021).
- [25] *Kanban: How the kanban methodology applies to software development*. URL: <https://www.atlassian.com/agile/kanban>. (accedido en: 14.03.2021).
- [26] Jake Knapp, John Zeratsky y Braden Kowitz. *The Design Sprint*. URL: <https://www.thesprintbook.com/how>. (accedido en: 21.02.2021).
- [27] Gaurav Kumar y Pradeep Bhatia. "Impact of Agile Methodology on Software Development Process". En: *International Journal of Computer Technology and Electronics Engineering (IJCTEE)* 2 (ago. de 2012), págs. 2249-6343.
- [28] Vitaly Kuprenko. *Heroku vs. AWS: Which Cloud Solution Works Best in 2020*. URL: <https://www.cloudacademy.com/blog/heroku-vs-aws-which-cloud-solution-works-best/>. (accedido en: 22.02.2021).
- [29] *Making a Pull Request*. URL: <https://www.atlassian.com/git/tutorials/making-a-pull-request>. (accedido en: 27.02.2021).
- [30] *Online communities and social communities: a primer*. URL: <https://www.i-scoop.eu/online-communities-social-communities-primer>. (accedido en: 26.02.2021).
- [31] *Qué es Plan Ceibal*. URL: <https://www.ceibal.edu.uy/es/institucional>. (accedido en: 10.04.2021).
- [32] M. Rizwan Qureshi. "Agile software development methodology for medium and large projects". En: *Software, IET* 6 (ago. de 2012), págs. 358-363. DOI: [10.1049/iet-sen.2011.0110](https://doi.org/10.1049/iet-sen.2011.0110).
- [33] Daniela Tomio y Andiara Paulta Hermann. "Mapeamento dos clubes de ciências da América Latina e construção do site da rede internacional de clubes de ciências". pt. En: *Ensaio Pesquisa em Educação em Ciências (Belo Horizonte)* 21 (2019). ISSN: 1983-2117. DOI: [10.1590/1983-21172019210111](https://doi.org/10.1590/1983-21172019210111). URL: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1983-21172019000100312&nrm=iso.
- [34] *What is Trello?* URL: <https://help.trello.com/article/708-what-is-trello>. (accedido en: 17.03.2021).

- [35] Colaboradores de Wikipedia. *Google Drive*. URL: https://en.wikipedia.org/wiki/Google_Drive. (accedido en: 17.03.2021).
- [36] Colaboradores de Wikipedia. *Google Hangouts*. URL: https://en.wikipedia.org/wiki/Google_Hangouts. (accedido en: 17.03.2021).
- [37] Colaboradores de Wikipedia. *JavaScript*. URL: es.wikipedia.org/wiki/JavaScript. (accedido en: 14.02.2021).
- [38] Colaboradores de Wikipedia. *MySQL*. URL: es.wikipedia.org/wiki/MySQL. (accedido en: 22.02.2021).
- [39] Colaboradores de Wikipedia. *Plataforma colaborativa*. URL: es.wikipedia.org/wiki/Plataforma_colaborativa. (accedido en: 15.10.2020).
- [40] Colaboradores de Wikipedia. *Slack*. URL: [https://en.wikipedia.org/wiki/Slack_\(software\)](https://en.wikipedia.org/wiki/Slack_(software)). (accedido en: 17.03.2021).

Anexos

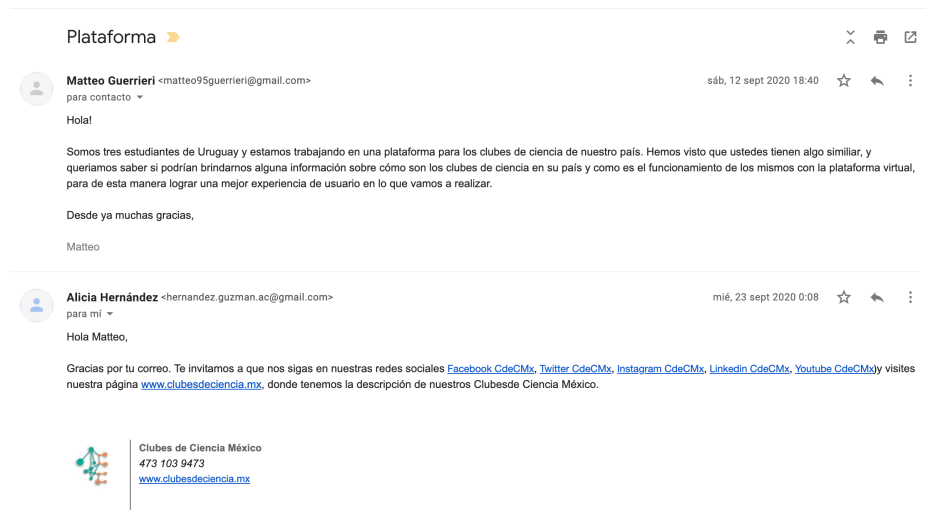


Figura 1: Email enviado a Clubes de Ciencia México

A. Diagrama de Afinidad

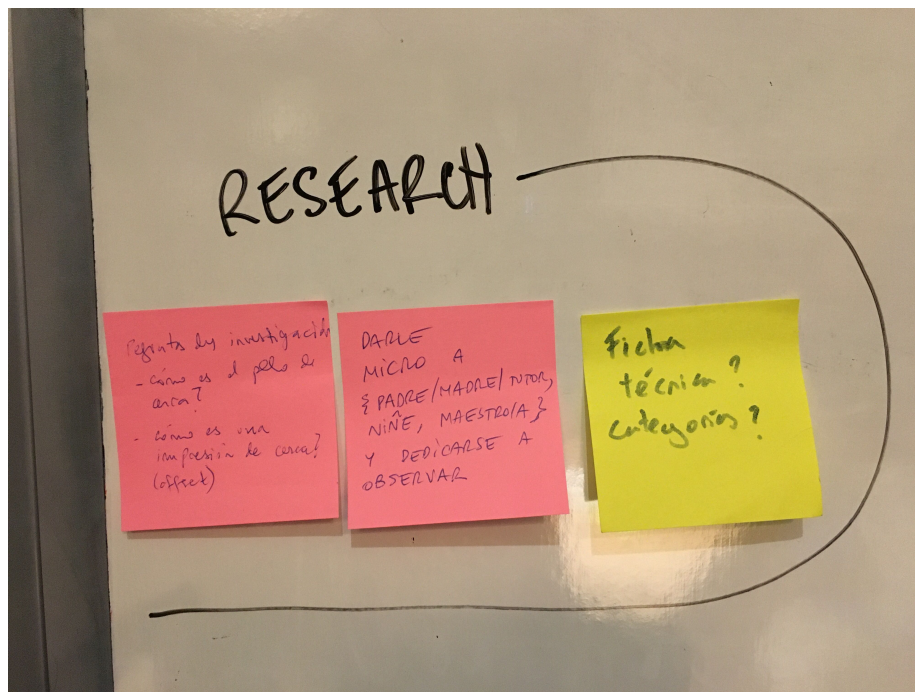


Figura A.1: Grupo Research

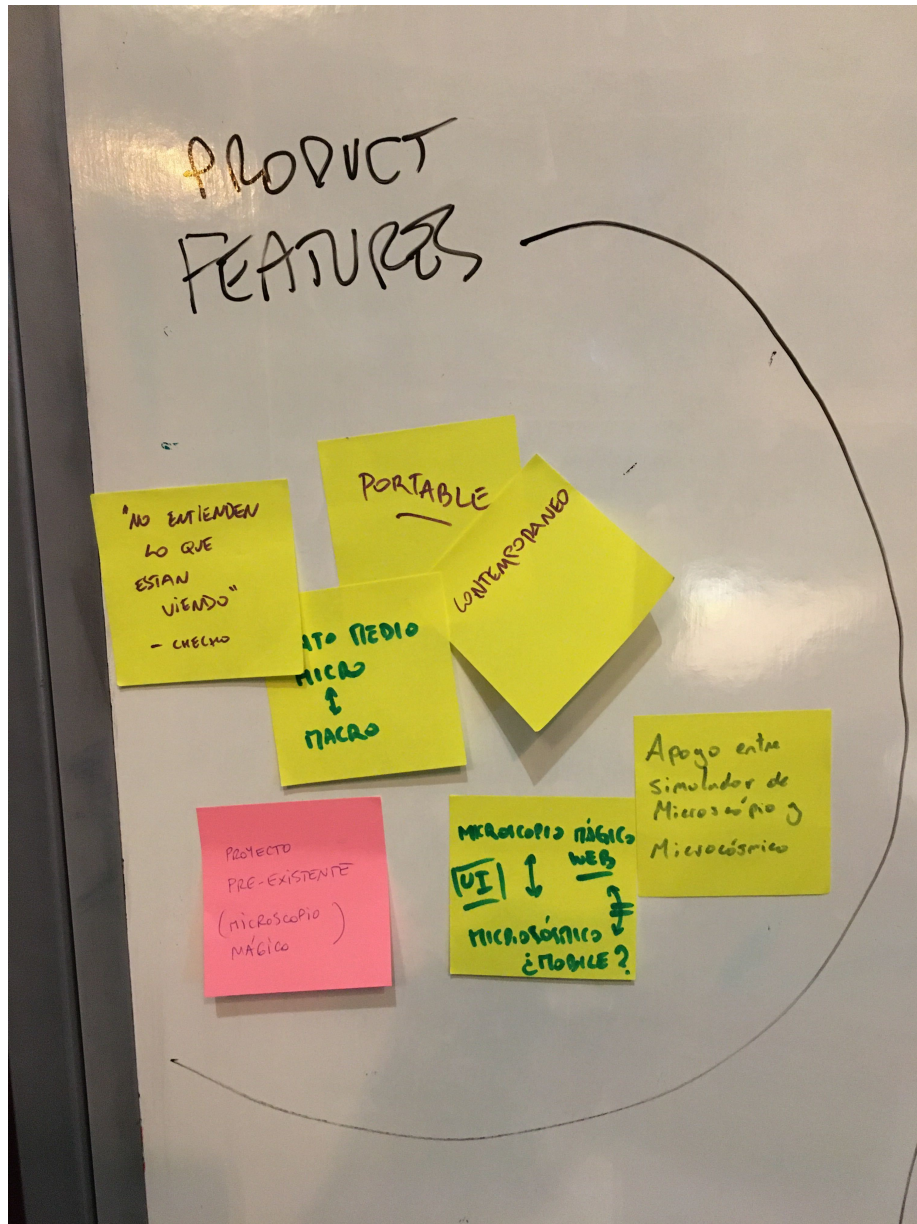


Figura A.2: Grupo Product Features

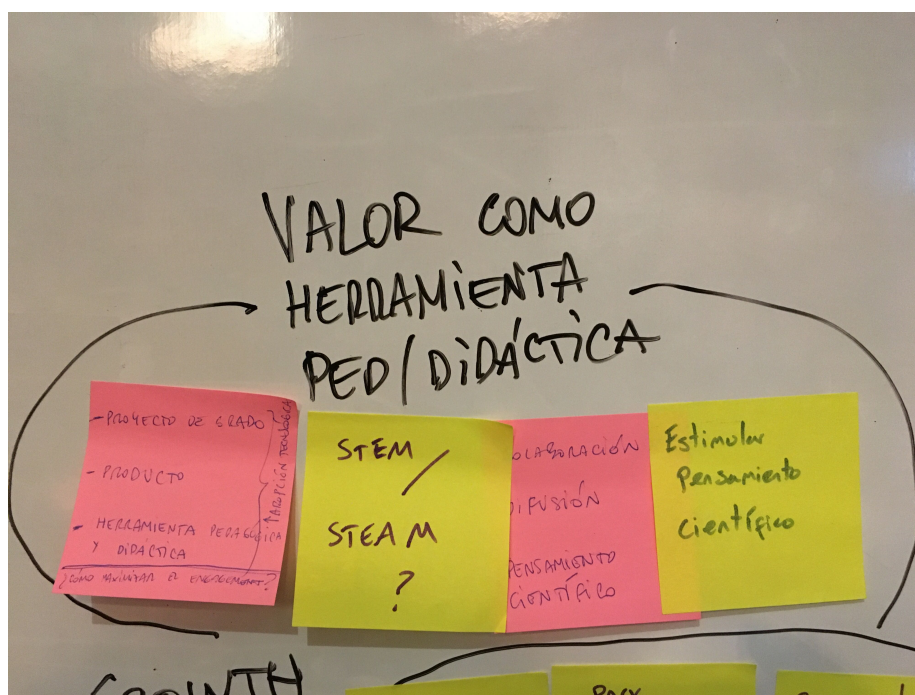


Figura A.3: Grupo Valor como herramienta

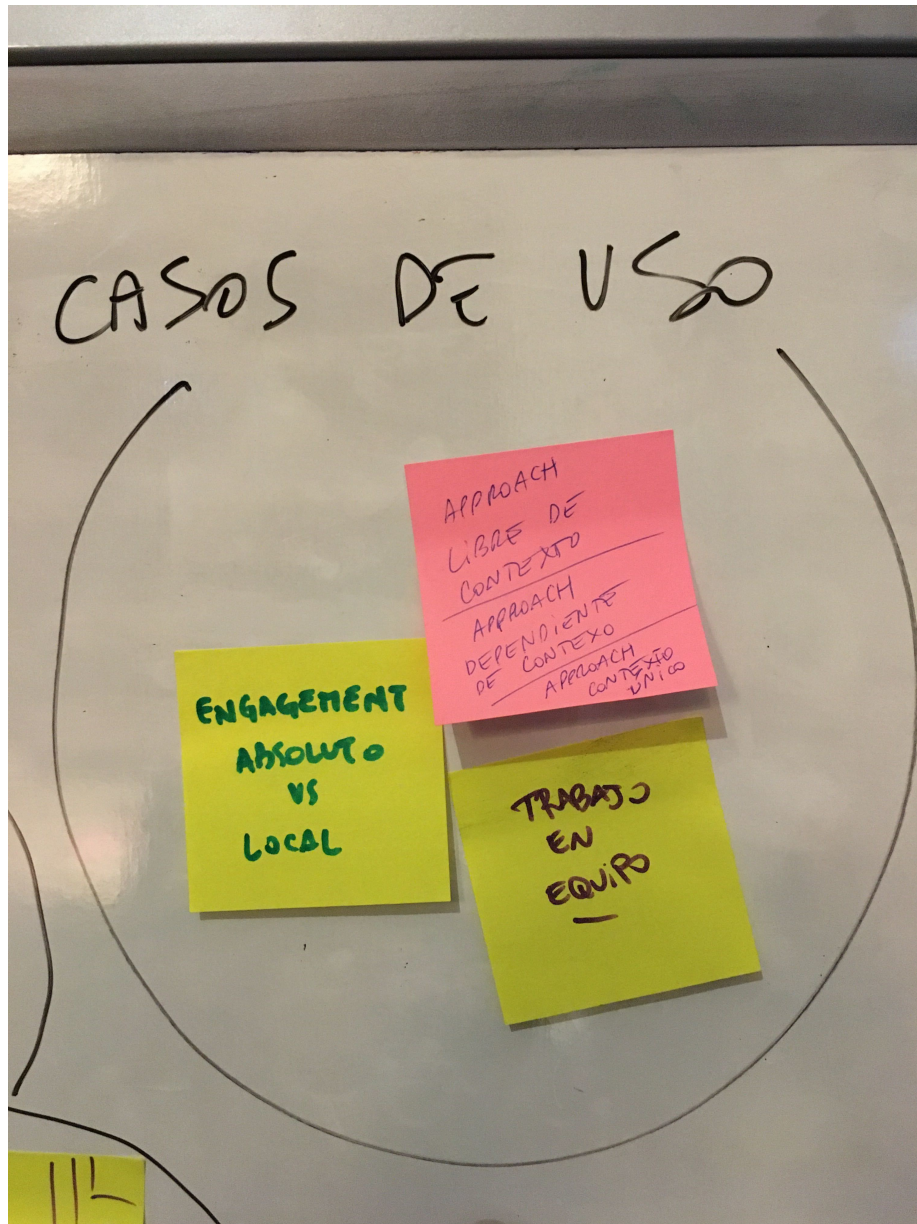


Figura A.4: Grupo Casos de Uso



Figura A.5: Grupo Growth Strategy

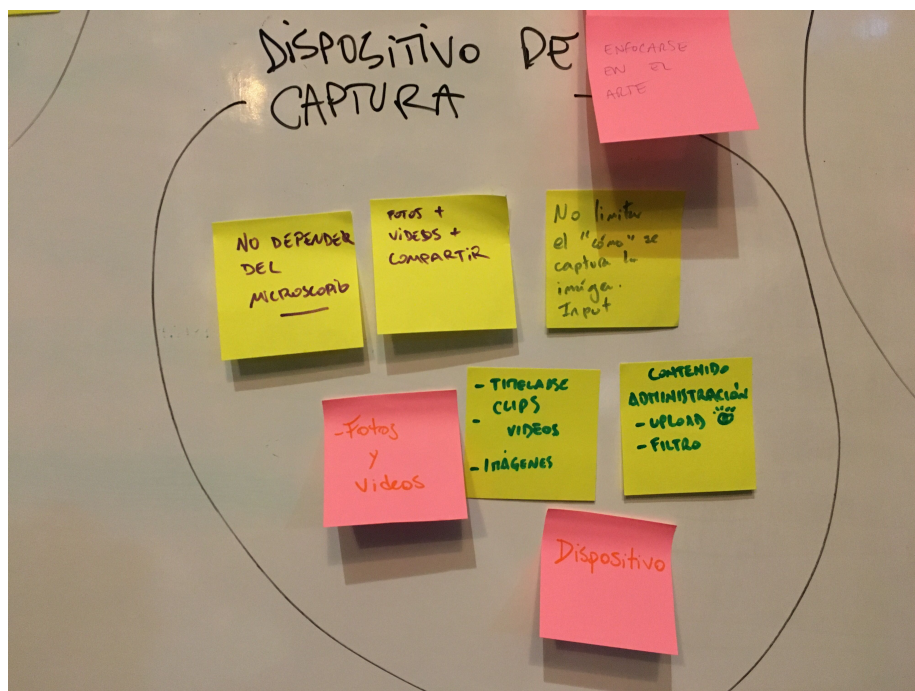


Figura A.6: Grupo Dispositivo de captura

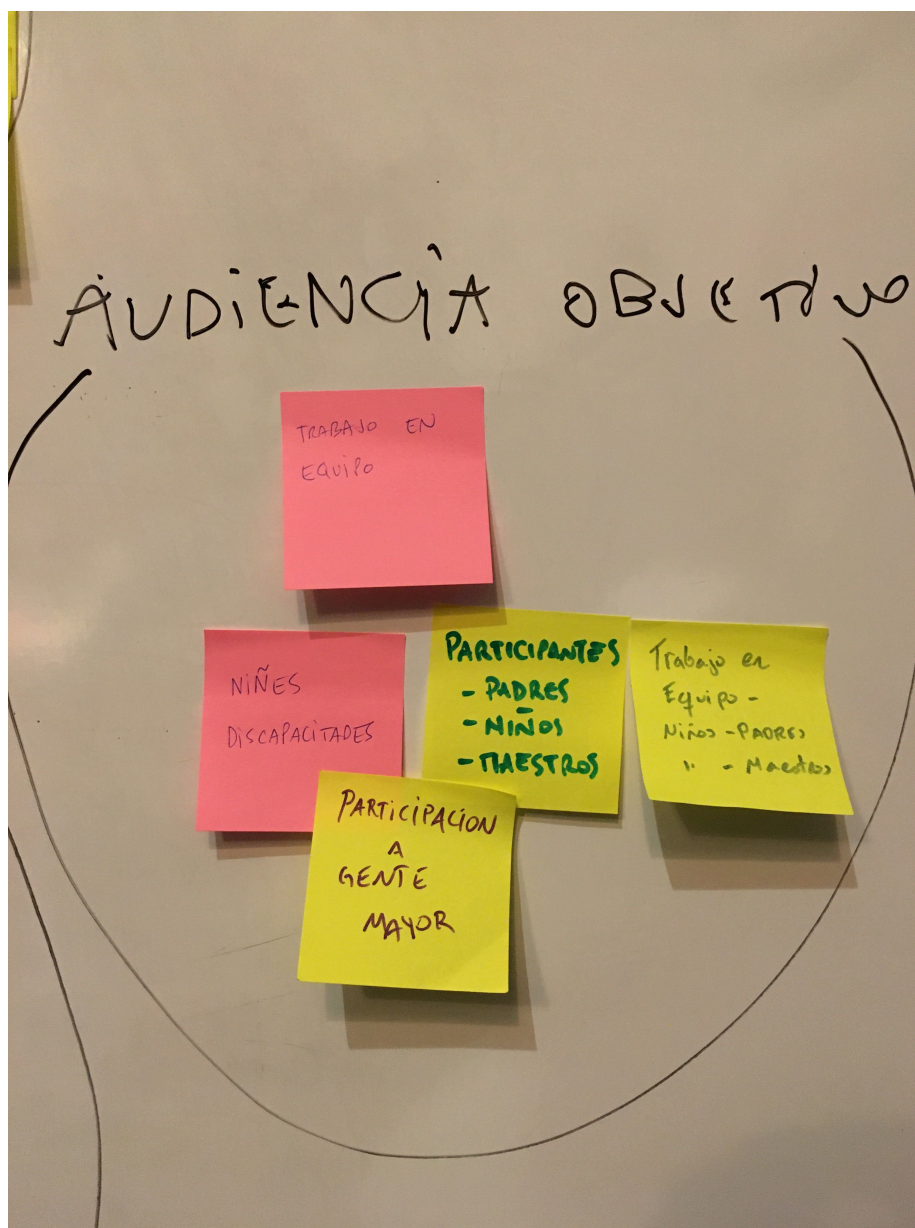


Figura A.7: Grupo Audiencia Objetivo

B. Instituto Clemente Estable

Nombre	Edad	Escuela? Pública/Privada	Tenés ceibalita?	Usas la ceibalita?	La usas? Para qué?	Alguna vez usaste un microscopio antes de venir acá?	Donde?	Qué viste?	Qué te gustaría ver?	Te gustaría tener uno?
Mateo	5	Privada / Maristas	No	No	No	Si	En su casa	Hormiga / Planta	Bichos	Ya tiene
Benjamin	7					No				
Valtín	8	Privada / British	No	Tiene computadora/ tablet		Si	En su casa			Ya tiene (No tiene pila)
Benjamin	8	Privada / JM	No	Tiene computadora	Si	No				Si
Alejo	8	Kenndy	No	Tiene computadora	Si	Si (De juguete)	En su casa			
Manuel	10	Maristas	No	Tiene computadora		Si	En su casa			
Pedro	8	Maristas	No	Tiene computadora		Si	En su casa			
Micaela	6					No				

Figura B.1: Entrevistas



Figura B.2: Experiencia en el Instituto



Figura B.3: Experiencia en el Instituto



Figura B.4: Experiencia en el Instituto

C. Crazy 8's

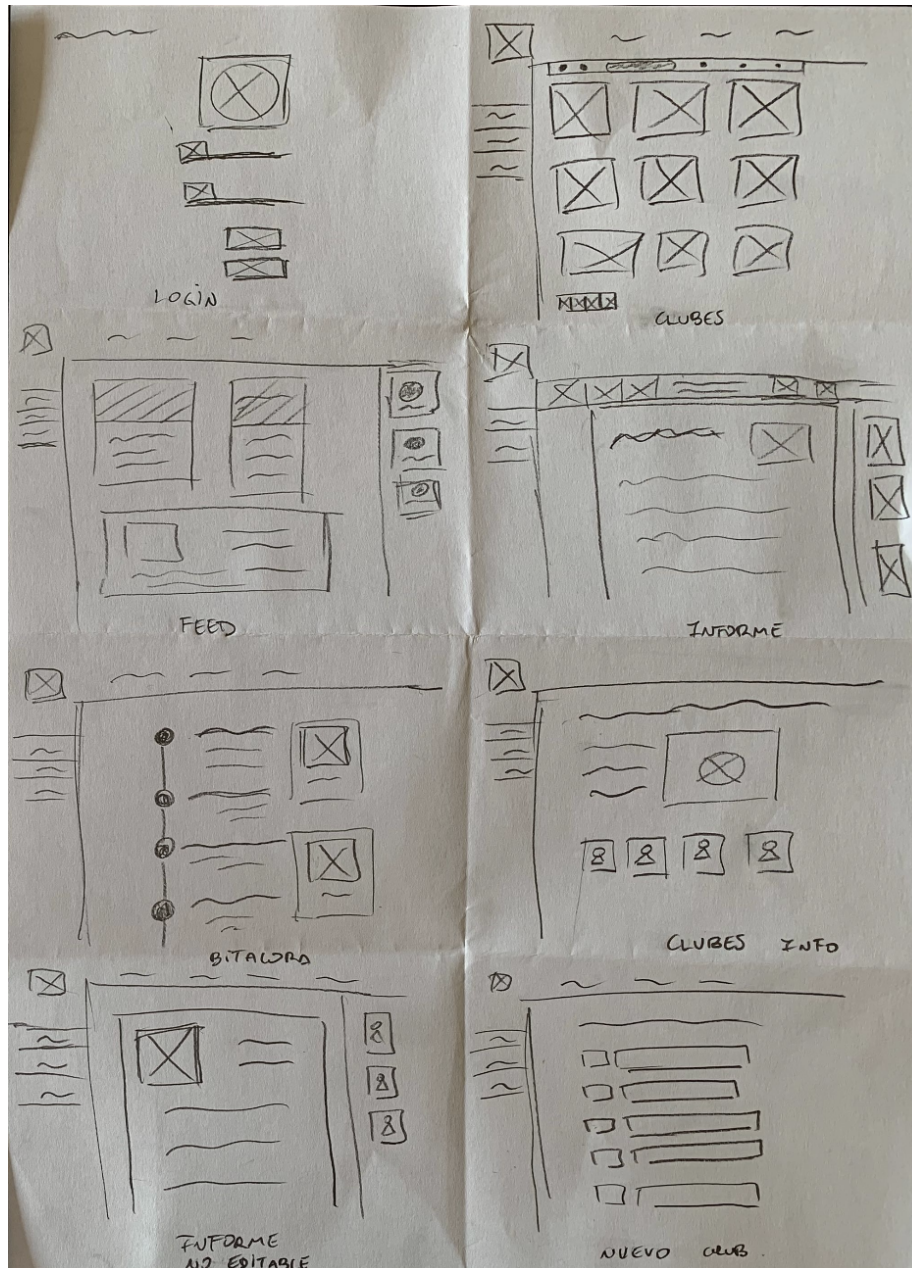


Figura C.1: Ejemplo de Crazy 8's

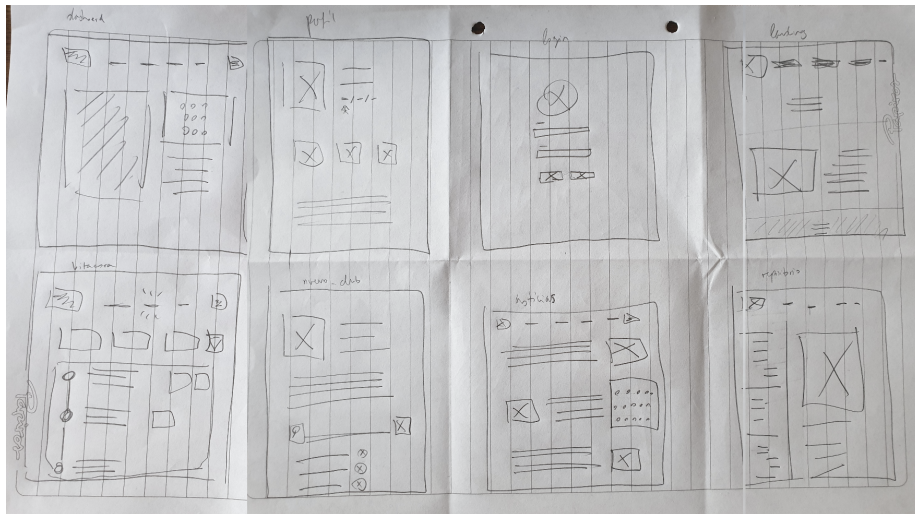


Figura C.2: Ejemplo de Crazy 8's

D. Historias de Usuario

Clubes de Ciencia - Historias de Usuario

	A	B	C	D	E
1	Como	quiero	para	Historia de Usuario	Observaciones
2	usuario	navegar la aplicación según cierto criterio		Como usuario quiero navegar la aplicación según cierto criterio	
3	"	que la aplicación se acomode a ciertos tamaños de pantalla		Como " quiero que la aplicación se acomode a ciertos tamaños de pantalla	
4	visitante	ver la landing principal		Como visitante quiero ver la landing principal	
5	visitante	registrarme	pasar a ser un usuario del sistema	Como visitante quiero registrarme, para pasar a ser un usuario del sistema	
6	visitante	hacer login	acceder a la aplicación web	Como visitante quiero hacer login, para acceder a la aplicación web	
7	usuario	hacer logout	pasar a ser un visitante del sistema	Como usuario quiero hacer logout, para pasar a ser un visitante del sistema	El sistema debe redirigir al usuario al onboarding si no tiene un Club de Ciencia activo
8	usuario	recibir una instrucción inicial básica (onboarding)		Como usuario quiero recibir una instrucción inicial básica (onboarding)	
9	usuario	ver la lista de Clubes de Ciencia agrupados según cierto criterio		Como usuario quiero ver la lista de Clubes de Ciencia agrupados según cierto criterio	
10	usuario	buscar Clubes de Ciencia por nombre, descripción, orientador		Como usuario quiero buscar Clubes de Ciencia por nombre, descripción, orientador	
11	usuario	seleccionar un ítem de la lista de Clubes de Ciencia	ver su detalle	Como usuario quiero seleccionar un ítem de la lista de Clubes de Ciencia, para ver su detalle	
12	usuario	marcar uno de mis Clubes de Ciencia como activo		Como usuario quiero marcar uno de mis Clubes de Ciencia como activo	
13	usuario	abrir el formulario de solicitud de membresía de un Club de Ciencia		Como usuario quiero abrir el formulario de solicitud de membresía de un Club de Ciencia	
14	usuario	enviar el formulario de solicitud de membresía de un Club de Ciencia		Como usuario quiero enviar el formulario de solicitud de membresía de un Club de Ciencia	
15	usuario	acceder a la pantalla principal de mi Club de Ciencia activo (home)		Como usuario quiero acceder a la pantalla principal de mi Club de Ciencia activo (home)	
16	usuario	acceder a mi Diario de Viaje desde la home		Como usuario quiero acceder a mi Diario de Viaje desde la home	
17	usuario	acceder a mi Libreta de Apuntes desde la home		Como usuario quiero acceder a mi Libreta de Apuntes desde la home	
18	usuario	acceder a la Carpeta de Campo desde la home		Como usuario quiero acceder a la Carpeta de Campo desde la home	
19	usuario	acceder al Póster desde la home		Como usuario quiero acceder al Póster desde la home	
20	usuario	acceder al Informe de Investigación desde la home		Como usuario quiero acceder al Informe de Investigación desde la home	
21					

Figura D.1: Historias de Usuario

E. Pruebas unitarias - RSpec

```
$ rspec spec --format documentation
```

Randomized with seed 52263

Missions

```
PATCH /update
  with valid parameters
    updates the requested mission
POST /create
  with valid parameters
    creates a new Mission
  with invalid parameters
    does not create a new Mission
request list of club missions
  returns all missions of a club
DELETE /destroy
  destroys the requested mission
```

FieldFolders

```
request one field folder
  returns field_folder by id
```

Clubs

```
request one club by id
  returns club by id
```

request list of all clubs
returns all clubs
destroy one club by id
returns club by id
update one club by id
updates club by id

Api::V1::ClubsController

routing
routes to *#destroy*
routes to *#index*
routes to *#create*
routes to *#update via PUT*
routes to *#show*

Entry

is valid with valid attributes
is not valid without a field_folder_id

Api::V1::EntriesController

routing
routes to *#show*
routes to *#create*
routes to *#update via PUT*
routes to *#destroy*
routes to *#index*

Api::V1::MissionsController

routing
routes to *#create*
routes to *#destroy*
routes to *#update via PUT*
routes to *#index*

FieldFolder

is not valid without a club_id
is valid with valid attributes

Club

is not valid without an category
is not valid without an area
is valid with valid attributes
is not valid without a name
when is not formal
 is not valid **if** category is hornero
when is formal
 is not valid **if** category is hornero

Mission

is valid with valid attributes
is not valid without a description

Api::V1::FieldFoldersController

routing
 routes to *#current*

Entries

update one entry by id
 updates entry by id
request list of all entries
 returns all entries of a field folder
request one entry by id
 returns entry by id
destroy one entry by id
 deletes an entry by id

Finished **in** 1.41 seconds (files took 2.24 seconds to load)
41 examples, 0 failures

Randomized with seed 52263

F. Pruebas E2E - Ghost Inspector

 Ghost Inspector

Upgrade Now!

Search...

Help

Matteo

[Dashboard](#) > [Fustibus](#)

Fustibus

Matteo Guerrieri – 7 Tests
Scheduled to run: [No Schedule](#)

[Settings](#) [More](#) [Run Entire Suite](#)

Tests

	TEST NAME	STATUS	SCREENSHOT	SCHEDULE	TESTED
☐	Crear entrada carpeta de campo ▶ Run Test ⚙ Edit Steps ⚙ Settings	🟢 PASSING	📷 PASSING	Suite Schedule 02/24/21 7:02 PM	
☐	Crear nueva nota ▶ Run Test ⚙ Edit Steps ⚙ Settings	🟢 PASSING	📷 PASSING	Suite Schedule 02/25/21 6:58 PM	
☐	Crear/Editar Poster ▶ Run Test ⚙ Edit Steps ⚙ Settings	🟢 PASSING	📷 PASSING	Suite Schedule 02/24/21 7:09 PM	
☐	Diario de Viaje ▶ Run Test ⚙ Edit Steps ⚙ Settings	🟢 PASSING	📷 PASSING	Suite Schedule 02/24/21 7:00 PM	
☐	Listado de clubes ▶ Run Test ⚙ Edit Steps ⚙ Settings	🟢 PASSING	📷 PASSING	Suite Schedule 02/24/21 6:57 PM	
☐	Show de Club ▶ Run Test ⚙ Edit Steps ⚙ Settings	🟢 PASSING	📷 PASSING	Suite Schedule 02/24/21 6:59 PM	
☐	Sign In ▶ Run Test ⚙ Edit Steps ⚙ Settings	🟢 PASSING	📷 PASSING	Suite Schedule 02/24/21 6:56 PM	

Create New Test

Test Name
ex: Add product to cart

Start URL
<https://staging.company.com>

Create Test

API Access

Your account includes API access. Below are some URLs for carrying out various actions. [Full documentation](#) →

- [Run this suite](#)
- [List results for this suite](#)
- [List all the tests in this suite](#)
- [Get details for this suite](#)

Figura F.1: Listado de pruebas realizadas

Ghost Inspector

[Upgrade Now!](#)

[Help](#)
Matteo

[Dashboard](#) > [Fustibus](#) > Crear entrada carpeta de campo

Crear entrada carpeta de campo
[Edit Steps](#)
[Settings](#)
[More](#)
[Run Test](#)

Matteo Guerrieri – Created by Matteo Guerrieri
 Scheduled to run: [Suite Schedule \(No Schedule\)](#)

Latest Completed Test Result
[Run Again with Result's Settings...](#)

Completed on Feb 24, 2021 @ 7:02:31 PM – Test duration was 0:33 – [Add Comment](#)
 Executed by Matteo Guerrieri with Chrome 88 at 1280x800 from Northern Virginia, USA
 Start URL: <https://fustibus.herokuapp.com/login>

Test Steps	PASSED
#1 Click on <code>input[name="email"]</code> ✓ URL: <code>https://fustibus.herokuapp.com/login</code> Time: 7:02:06 PM Edit	
#2 Assign <code>juanprueba@mail.com</code> into <code>input[name="email"]</code> ✓ URL: <code>https://fustibus.herokuapp.com/login</code> Time: 7:02:08 PM Edit	
#3 Assign <code>*****</code> into <code>input[name="password"]</code> ✓ URL: <code>https://fustibus.herokuapp.com/login</code> Time: 7:02:09 PM Edit	
#4 Click on <code>//button[contains(text(), "Ingresar")]</code> ✓ URL: <code>https://fustibus.herokuapp.com/login</code> Time: 7:02:10 PM Edit	
#5 Click on <code>//button[contains(text(), "Carpeta de Campo")]</code> ✓ URL: <code>https://fustibus.herokuapp.com/</code> Time: 7:02:11 PM Edit	
#6 Click on <code>button[type="button"]</code>	

Video Recording

Screenshot – 0.00% Change

Figura F.2: Crear entrada en carpeta de campo

Ghost Inspector

[Upgrade Now!](#)

[Help](#)
Matteo

[Dashboard](#) > [Fustibus](#) > Crear nueva nota

Crear nueva nota
[Edit Steps](#)
[Settings](#)
[More](#)
[Run Test](#)

Matteo Guerrieri – Created by Matteo Guerrieri
 Scheduled to run: [Suite Schedule \(No Schedule\)](#)

Latest Completed Test Result
[Run Again with Result's Settings...](#)

Completed on Feb 25, 2021 @ 6:58:21 PM – Test duration was 0:20 – [Add Comment](#)
 Executed by Matteo Guerrieri with Chrome 88 at 1280x800 from Northern Virginia, USA
 Start URL: <https://fustibus.herokuapp.com/login>

Test Steps	PASSED
#1 Click on <code>input[name="email"]</code> ✓ URL: <code>https://fustibus.herokuapp.com/login</code> Time: 6:58:07 PM Edit	
#2 Assign <code>juanprueba@mail.com</code> into <code>input[name="email"]</code> ✓ URL: <code>https://fustibus.herokuapp.com/login</code> Time: 6:58:08 PM Edit	
#3 Assign <code>*****</code> into <code>input[name="password"]</code> ✓ URL: <code>https://fustibus.herokuapp.com/login</code> Time: 6:58:09 PM Edit	
#4 Click on <code>//button[contains(text(), "Ingresar")]</code> ✓ URL: <code>https://fustibus.herokuapp.com/login</code> Time: 6:58:10 PM Edit	
#5 Click on <code>//button[contains(text(), "Libreta de Apuntes")]</code> ✓ URL: <code>https://fustibus.herokuapp.com/</code> Time: 6:58:11 PM Edit	
#6 <code>css=tbody contains text Libreta de apuntes</code>	

Video Recording

Screenshot – 0.00% Change

Figura F.3: Crear nueva nota

 Ghost Inspector

Upgrade Now!

Search...

Help

Matteo

[Dashboard](#) > [Fustibus](#) > Show de Club

Show de Club

Matteo Guerrieri – Created by Matteo Guerrieri
Scheduled to run: [Suite Schedule \(No Schedule\)](#)

[Edit Steps](#) [Settings](#) [More](#) [Run Test](#)

Latest Completed Test Result

Completed on Feb 24, 2021 @ 6:59:18 PM – Test duration was 0:24 – [Add Comment](#)
Executed by Matteo Guerrieri with Chrome 88 at 1280x800 from Northern Virginia, USA
Start URL: <https://fustibus.herokuapp.com/login>

[Run Again with Result's Settings...](#)

Test Steps	PASSED
#1 Click on <code>input[name="email"]</code> URL: https://fustibus.herokuapp.com/login Time: 6:58:59 PM Edit	
#2 Assign <code>juangrueba@gmail.com</code> into <code>input[name="email"]</code> URL: https://fustibus.herokuapp.com/login Time: 6:59:00 PM Edit	
#3 Assign <code>*****</code> into <code>input[name="password"]</code> URL: https://fustibus.herokuapp.com/login Time: 6:59:01 PM Edit	
#4 Click on <code>//button[contains(text(), "Ingresar")]</code> URL: https://fustibus.herokuapp.com/login Time: 6:59:01 PM Edit	
#5 Click on <code>#menubutton-4</code> URL: https://fustibus.herokuapp.com/ Time: 6:59:03 PM Edit	
#6 Click on <code>//button[contains(text(), "Mis clubes")]</code>	

Video Recording



Screenshot – 0.00% Change [View](#) **PASSED**



Figura F.4: Show de un club