

Duarte, David Julian

Práctica Profesional Supervisada

Instituto: Ingeniería y Agronomía

Carrera: Ingeniería en Informática



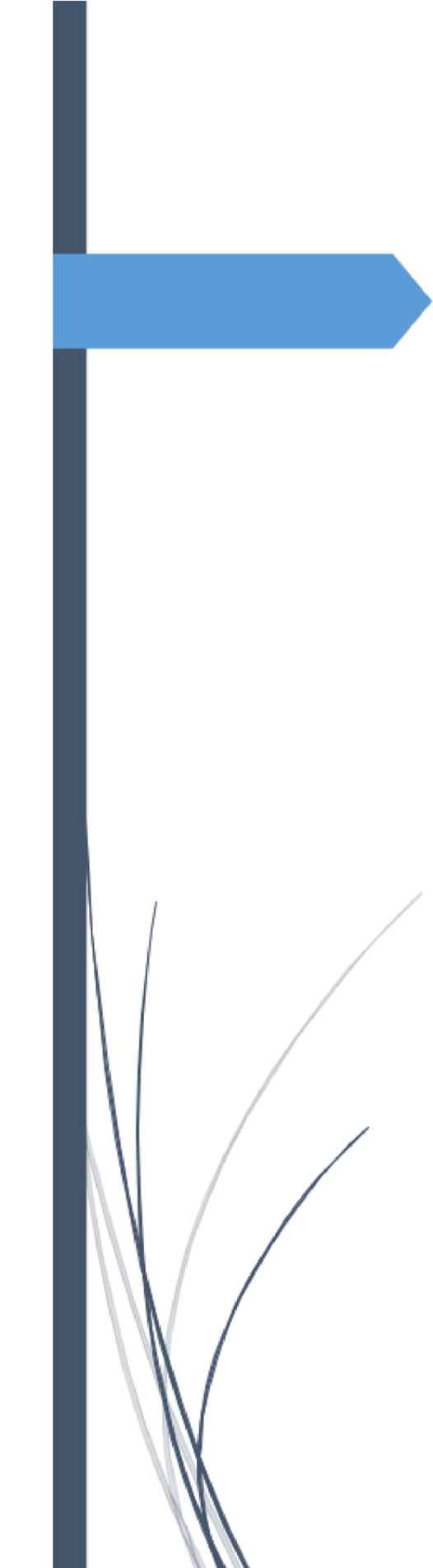
Esta obra está bajo una Licencia Creative Commons Argentina.
Atribución – no comercial 4.0
<https://creativecommons.org/licenses/by-nc/4.0/>

Documento descargado de RID - UNAJ Repositorio Institucional Digital de la Universidad Nacional Arturo Jauretche

Cita recomendada:

Duarte, D.J. *Práctica Profesional Supervisada* [Informe de la práctica Profesional Supervisada] Universidad Nacional Arturo Jauretche

Disponible en RID - UNAJ Repositorio Institucional Digital UNAJ <https://biblioteca.unaj.edu.ar/rid-unaj-repositorio-institucional-digital-unaj>



Práctica Profesional Supervisada

Informe final

David Julián Duarte

legajo 3994

UNIVERSIDAD NACIONAL ARTURO JAURETCHE

INSTITUTO DE INGENIERIA Y AGRONOMIA

INGENIERIA INFORMATICA

PRÁCTICA PROFESIONAL SUPERVISADA (PPS)
Informe Plan de Trabajo Preliminar

DATOS DEL ESTUDIANTE

| | |
|---|--|
| Apellido y Nombres: | Duarte, David Julián |
| Nº de Legajo: | 3994 |
| Correo electrónico: | davidjulianduarte@gmail.com |
| Cantidad de materias aprobadas al comienzo de la PPS: | 43 |

PROFESOR TUTOR DE LA UNAJ

| | |
|---------------------|--|
| Apellido y Nombres: | Dr. Ing. Morales, Martin |
| Correo electrónico: | martin.morales@unaj.edu.ar |

**PROFESOR TUTOR DEL TALLER DE APOYO A LA PRODUCCIÓN DE TEXTOS
ACADÉMICOS DE LA UNAJ**

| | |
|---------------------|--|
| Apellido y Nombres: | Bein, Paula Mariana |
| Correo electrónico: | paula.bein@gmail.com |

DATOS DE LA ORGANIZACIÓN DONDE SE REALIZA LA PPS

| | |
|------------------------|----------------------------|
| Nombre o Razón Social: | FluxIT S.A. |
| Dirección: | calle 9 N 865 PA, La Plata |
| Teléfono: | 0221 445 3480 |
| Sector: | Herramientas Internas |

TUTOR DE LA EMPRESA/INSTITUCIÓN

| | |
|---------------------|--|
| Apellido y Nombres: | Iaria, Pablo |
| Correo electrónico: | iariap@fluxit.com.ar |

FIRMA DEL COORDINADOR DE LA CARRERA

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

Índice

| | |
|--|-----------|
| 1. Introducción | 4 |
| 2. Temas de estudio | 6 |
| 2.1. Beneficios Sociales | 6 |
| 2.2. Aplicaciones Web Progresivas (PWA) | 8 |
| 2.2.1. Tecnologías que usan las PWA | 8 |
| 2.2.2. Características de las PWA | 9 |
| 2.2.3. Ventajas de las PWA | 11 |
| 2.2.4. Desventajas de las PWA | 12 |
| 2.2.5. Elección PWA | 12 |
| 3. Herramientas seleccionadas | 12 |
| 3.1. Vue.js | 13 |
| 3.1.1. Vue.js en comparación con otros Frameworks js | 13 |
| React | 14 |
| Angular 1 | 16 |
| Angular 2 | 16 |
| Ember | 17 |
| Polymer | 18 |
| Riot | 19 |
| 3.1.2. Elección de Vue.js | 19 |
| 3.2. Firebase | 19 |
| 3.2.1. Ventajas y Desventajas | 20 |
| 3.2.2. Realtime Database | 21 |
| 3.2.3. Firebase Storage | 22 |
| 3.2.4. Firebase Auth | 22 |

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

| | |
|--|-----------|
| 4. Diseño, desarrollo e implementación de un sistema aplicado al control de beneficios sociales en FluxIT | 23 |
| 4.1. 1º Sprint | 26 |
| 4.2. 2º Sprint | 31 |
| 4.3. 3º Sprint | 35 |
| 4.4. 4º Sprint | 36 |
| 4.5. 5º Sprint | 40 |
| 4.6. 6º Sprint | 44 |
| 5. Conclusión: Problemas generales surgidos al realizar la aplicación | 45 |
| 6. Bibliografía utilizada | 47 |

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

1. Introducción

El presente trabajo mostrará el diseño, desarrollo e implementación de un sistema web para el control de méritos personales en la empresa FluxIT S.A. con la finalidad de brindar una retribución, o beneficios, a los trabajadores que desarrollen una labor eficaz, un mayor esfuerzo o bien, que manifiesten un compromiso con la empresa. También se considerará la participación y rendimiento en capacitaciones.

La aplicación se sumará a una ya existente de retroalimentación.

Para el desarrollo de la aplicación se utilizarán tecnologías de desarrollo web: HTML, CSS y librerías Javascript. Para el almacenamiento de los datos se utilizará Firebase, una tecnología que permite realizar aplicaciones web sin la necesidad de montar un servidor local.

Se tomará *Scrum* como modelo de desarrollo de software y como herramienta complementaria para el manejo de tareas, *Trello*, un software de administración de proyectos con interfaz web.

Se espera que la aplicación permita al personal de Valor Humano recompensar la labor de los trabajadores (*Fluxers*)¹ a través de diferentes premios. Para ello debe permitir dos roles de usuarios: un usuario general y un usuario administrador.

Cada *Fluxer* tendrá un usuario general que permitirá la gestión de premios, poder ver cómo conseguir *recoins*², enviarlos a través del sistema de *Feedback*³ y acceder a los movimientos del mismo.

El personal de Valor Humano tendrá un usuario administrador que le permitirá la gestión de los premios y de las tareas.

¹ De acá en adelante el Fluxer es la persona que trabaja en cualquier área de FluxIT.

² Recoins es la moneda ficticia que utilizará el sistema.

³ El sistema de Feedback, es una aplicación web progresiva por la cual los Fluxers se hacen retribuciones entre ellos en pos de registrar críticas constructivas.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

La aplicación de este sistema permitirá fomentar la participación de los *Fluxers* en actividades voluntarias como la participación en la radio de FluxIT, publicar en la revista institucional o mismo retribuir un buen trabajo.⁴

También brindará una nueva herramienta a Valor Humano en el labor de la contención del personal. Por contención se entiende a mantener motivados a los trabajadores, brindar un óptimo seguimiento personal, garantizar su valor en la empresa y evitar su deserción.

2. Temas de estudio

Con el objetivo de comprender la realización de la aplicación se plantea estudiar dos temas principales:

- 1) Primero se van a detallar los beneficios sociales. La aplicación se encargará de brindar soporte a los mismos. Es de utilidad comprender de qué se tratan, qué son, algunos ejemplos, los objetivos personales y para la empresa.
- 2) Luego se tratarán las aplicaciones progresivas, tecnología seleccionada para dar respuesta a la gestión de beneficios sociales en FluxIT.

2.1. Beneficios Sociales

La aplicación proveerá una solución en la administración de los beneficios sociales y permitirá una retribución equitativa en referencia a la participación y

⁴ FluxIT cuenta con una edición mensual de una radio y una revista interna en la cual se dan novedades de la empresa y entrevistas a Fluxers. En la radio también brindan música y noticias del ámbito informático y en la revista se publican artículos especializados escritos por alguno de los Fluxers.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

esfuerzo de los Fluxers que persigan continuar formándose profesionalmente, compartir sus conocimientos, al mismo tiempo que garantizar un software de calidad.

2.1.1. ¿Qué son los Beneficios Sociales?

Las empresas buscan que los trabajadores se encuentren motivados, es por ello que persiguen satisfacer las necesidades de los mismos.

Se considera 'beneficios' a aquellas regalías y ventajas que las organizaciones conceden a la totalidad o parte de los empleados como retribuciones adicionales de los salarios. Incluyen variedad de comodidades y ventajas ofrecidas por la organización, como transporte, pago de tiempo no laborado, descuentos en capacitaciones, material bibliográfico y premios de producción. También incluyen descuentos en restaurantes, cafeterías y alimentación subsidiada y, tiempo libre, entre otros.

Los objetivos de estos surgen de la preocupación de mantener al personal y reducir la rotación.

2.1.2. Objetivos de los beneficios sociales

Como ya se ha dicho, los beneficios tratan de atender las necesidades individuales de las personas, proporcionándoles una vida personal, familiar y laboral más tranquila y productiva. Tratan de suministrarles condiciones para que cada persona se pueda desligar de las preocupaciones diarias y concentrar en las actividades laborales, para satisfacer necesidades más elevadas.

La base de los mismos es crear un ambiente que resulte, por un lado, atractivo para el trabajador y, al mismo tiempo, se pueda evitar la rotación permanente del personal. En este sentido, se busca:

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

- Reducir la fatiga física y psicológica de las personas.
- Apoyar el reclutamiento de personal y atraer candidatos.
- Reducir la rotación del personal y promover la permanencia de las personas en la empresa.
- Reducir el ausentismo.
- Minimizar el costo de las horas trabajadas.

Además, permiten a los trabajadores satisfacer necesidades complejas, tales como afiliación, reconocimiento e incluso autorrealización. Tomando la pirámide de Maslow como referencia, estas tres, son las jerarquías más altas.

Si se consigue satisfacer estas necesidades, se logrará un mejor ambiente laboral.

2.2. Aplicaciones Web Progresivas (PWA)

Las aplicaciones web progresivas son páginas web que se comportan como aplicaciones nativas. Se accede por una URL, que es un navegador web y se adapta al formato de la pantalla para que simule el entorno de una aplicación. Como se encuentran disponibles a través de la web, no es necesario que el usuario instale o descargue nada, las mismas no están disponibles en el mercado de aplicaciones (Play Store, App Store, Microsoft Store).

Otra de las ventajas principales de las aplicaciones web progresivas es que se cargan rápidamente, incluso con redes débiles. Además permite enviar notificaciones push relevantes y contener un ícono de acceso rápido en la pantalla principal del dispositivo. Se construyen utilizando estándares de desarrollo web, algunos ya conocidos como HTML, CSS y JavaScript; y una nueva generación de APIs de JavaScript.

2.2.1. Tecnologías que usan las PWA

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

Las aplicaciones web progresivas hacen uso de las últimas tecnologías desarrolladas para los navegadores modernos. Algunas de éstas que se tuvieron en cuenta para el presente trabajo son las siguientes:

- Responsive Web Design: permite adaptar páginas web a diversos dispositivos que poseen diferentes tamaños de pantalla.
- Service Workers: para ejecutar servicios en segundo plano en los navegadores. Son independientes a la aplicación (es decir, funcionan incluso si la aplicación está cerrada). Ofrecen capacidades avanzadas como la intercepción de las comunicaciones, el cacheado de información, la descarga en segundo plano de contenidos, el trabajo sin conexión o la posibilidad de enviar notificaciones.
- App Shell: es un patrón de desarrollo que consiste en separar la aplicación entre funcionalidad y contenido para poder cargar ambos por separado. De esta forma, se puede mejorar el tiempo de respuesta. Lo suyo es tener, por un lado, la aplicación en sí cacheada para uso off-line (con Service Workers o no) de modo que la página esté lista rápidamente, y luego el contenido (los datos) que cargue por otro lado, inicialmente desde una caché y luego se actualicen, o directamente desde la web si hay conexión. Esto, bien realizado, hace que la percepción que tiene el usuario de la velocidad de carga de la app sea mayor. Parece que es mucho más rápido porque al cargar el "shell" antes de nada y desde una caché, el usuario verá la app enseguida.
- Manifiesto de aplicación: es un archivo que sirve para poder cambiar la configuración de la aplicación Android

2.2.2. Características de las PWA

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

Se detallan las características principales de las aplicaciones web progresivas:

- Son seguras. Uso de HTTPS para mejorar la seguridad que ofrecen las aplicaciones web estandarizadas.
- Son progresivas. Funcional para cada usuario, sin importar el navegador pues su principio es la mejora progresiva.
- De diseño responsive. Disponible para ordenador, tablet, móvil o cualquier otro dispositivo.
- Funcionan independientemente de la conectividad. Capacidad de trabajar sin conexión o con poca conectividad.
- Se encuentran actualizadas. Siempre se mantiene actualizada gracias a los Service Workers.
- Son como apps. Tanto su aspecto, navegación e interacciones son comparables con las de una app nativa.
- Son localizables: Se identifica como aplicaciones debido a que pueden ser encontradas por motores de búsqueda.
- Permiten acciones de re-engagement. Mantiene el interés y compromiso de los usuarios mediante características como notificaciones
- Son enlazables. Se pueden compartir fácilmente mediante su URL.

De las características principales de las aplicaciones web progresivas mencionadas anteriormente, se separan las siguientes con el objetivo de hacer un análisis más profundo. Las mismas están más ligadas al desarrollo mobile que al desarrollo web y son de suma importancia para el proyecto:

- Push Notifications. Las notificaciones en el frontend no son nuevas; existen muchas aplicaciones que las usan, desde Inbox, Facebook Messenger, Whatsapp, hasta Google Play Music y más. Estas notificaciones se generan desde javascript y son muy sencillas de crear.

La novedad es que estas notificaciones ahora se pueden recibir y mostrar en el background, sin la necesidad de tener abierta la página de la que en ese momento se reciben las notificaciones. Facebook utiliza esta técnica y

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

permite recibir notificaciones en el background.

Otro ejemplo es ProductHunt, una página donde se pueden ver startups y productos nuevos. Cada día, envían notificaciones de dichos productos que se reciben sin la necesidad de instalar nada en la computadora, ni abrir o visitar la página.

- Página en Home Screen. Este feature se hace mediante el uso de un manifiesto que se coloca en la carpeta principal de la página web. Éste indica al dispositivo cómo debe agregar el icono al home de la pantalla. Mientras el usuario está en la aplicación se le notifica de la posibilidad de agregar un acceso directo “Add to Home Screen”. Si se presiona el botón, se le crea un icono en el home a través del cual acceder a la página web sin necesidad de instalar ninguna aplicación.

2.2.3. Ventajas de las PWA

Sumado a las características que se mencionaron previamente como la seguridad, el diseño responsivo y su modo offline; se mencionan algunas ventajas más en comparación con aplicaciones nativas y web:

- **Velocidad:** Las aplicaciones web progresivas ofrecen una interacción rápida y responsiva, cercana o igual a la experiencia de las aplicaciones nativas. Velocidad significa muchas cosas, un render rápido, animaciones fluidas, responder rápido a los eventos del navegador, transiciones, no recargar la página, etc.
Las aplicaciones web progresivas pueden, incluso, ser más rápidas que las nativas, porque corren en una aplicación que el usuario tiene normalmente abierta, el navegador. A diferencia de apps nativas que normalmente no están abiertas.
- **Distribución:** como están alojadas en un servidor el usuario no tiene que instalar nada. Además, esto ofrece importantes ventajas a la hora de distribuir actualizaciones garantizando que todos los usuarios utilicen la

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

misma versión de la aplicación. Los usuarios reciben las actualizaciones de forma instantánea, debido a que la aplicación se modifica en un único punto y no es necesario realizar el proceso a través de las app store.

- Costos: El desarrollo de Apps nativas es costoso. Requieren de licenciamiento, de aprobación de la aplicación y el desarrollador debe dejar el 30% de sus ganancias como tributo a Apple o Google por usar dichas tiendas. Sumado al tiempo de desarrollo de uno o varios programadores para crear una aplicación multiplataforma. Las aplicaciones web progresivas no pasan por la tienda por lo que no se debe pagar licenciamiento y el desarrollo es para una sola plataforma, lo que produce costos menores.

2.2.4. Desventajas de las PWA

En comparación con las aplicaciones nativas se mencionan algunas importantes desventajas:

- No todos los exploradores y sistemas operativos soportan aplicaciones progresivas. Siendo iOS un sistema operativo que todavía no las soporta.
- No se pueden usar todas las funcionalidades nativas del dispositivo. Entre estas estarían, por ejemplo, los contactos, el calendario, bluetooth o NFC.

2.2.5. Elección PWA

Se eligió realizar una aplicación web progresiva porque, sumado a las ventajas antes mencionadas, brindan una solución liviana y económica a la necesidad de una aplicación multi plataformas.

Si bien con la elección de esta tecnología no alcanzamos el mercado iOS en su totalidad, se plantea resolver las notificaciones importantes con el envío de correos electrónicos.

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|-------------------------|----------------------|

3. Herramientas seleccionadas

Primeramente se detallaron los beneficios sociales y su importancia en una organización. En pos de mejorar la administración de los mismos se decidió el desarrollo de una aplicación web progresiva. Anteriormente, se presentaron las características de estas aplicaciones y se mencionó que hacen uso de una nueva generación de APIs de JavaScript.

Esta aplicación, complementaria a la de Feedback y que forma parte de una ampliación de esta misma, hace uso de Vue.js, una API perteneciente a la nueva generación de JavaScript que permite crear aplicaciones progresivas. A su vez, también utiliza Firebase y Moment.js entre otros.

Firebase permite tener la base de datos en un ambiente Cloud y desprenderse de la administración de un servidor. Cuenta con la seguridad, los servicios y los servidores de Google Cloud.

Se detalla a continuación más información sobre estas aplicaciones y servicios.

3.1. Vue.js

Vue.js es una librería javascript progresiva y open-source para construir interfaces web interactivas, la documentación dice que no es un framework porque sólo se especializa en la capa de vista (VM - ViewModel).

A diferencia de otros frameworks monolíticos, Vue está diseñado desde el inicio para ser adoptado incrementalmente. La biblioteca principal se enfoca sólo en la capa de la vista, y es muy simple de utilizar e integrar con otros proyectos o bibliotecas existentes. Por otro lado, Vue también es perfectamente capaz de soportar aplicaciones sofisticadas de una sola página (en inglés single-page-application o SPA) cuando se utiliza en combinación con herramientas modernas y librerías compatibles.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

3.1.1. Vue.js en comparación con otros Frameworks js

Siendo una librería relativamente nueva (2014), y en un mercado en el que abundan los frameworks Javascript, se menciona una comparativa realizada por los desarrolladores de Vue contra los más utilizados en el mercado.

React

React y Vue tienen varias similitudes:

- Utilizan un DOM virtual
- Proveen componentes de vista reactiva
- Mantienen el foco en la biblioteca central, con preocupaciones tales como el enrutamiento y la administración de estado global manejada por las bibliotecas complementarias

Rendimiento:

Cuando se procesa la interfaz de usuario, la manipulación del DOM suele ser la operación más costosa y ninguna biblioteca puede hacer que esas operaciones crudas sean más rápidas. Para reducir el procesamiento se puede:

- Minimizar el número de mutaciones de DOM necesarias. Tanto React como Vue usan abstracciones DOM virtuales para lograr esto y ambas implementaciones funcionan igual de bien.
- Agregar la menor sobrecarga posible (cálculos puros de JavaScript) además de las manipulaciones DOM. Esta es un área donde Vue y React difieren.

La sobrecarga de JavaScript está directamente relacionada con los mecanismos de cálculo de las operaciones DOM necesarias. Tanto Vue como React utilizan DOM virtual para lograr eso, pero la implementación Virtual DOM de Vue es mucho más liviana y por lo tanto presenta menos sobrecarga que React.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

Tanto Vue como React también ofrecen componentes funcionales, que son sin estado y sin instancia, y por lo tanto, requieren menos sobrecarga. Cuando se usan en situaciones de rendimiento crítico, Vue es más rápido. Para demostrar esto en la figura 1 se observa un proyecto de referencia simple que sólo procesa 10.000 elementos de lista 100 veces.

| | Vue | React |
|------------|-------|-------|
| Fastest | 23ms | 63ms |
| Median | 42ms | 81ms |
| Average | 51ms | 94ms |
| 95th Perc. | 73ms | 164ms |
| Slowest | 343ms | 453ms |

Figura 1. Comparativa rendimiento React y Vue

Actualizar Rendimiento:

En React, cuando el estado de un componente cambia, desencadena la repetición de todo el subárbol del mismo, comenzando en ese como raíz. Para evitar repeticiones innecesarias de componentes secundarios, React debe implementar *shouldComponentUpdate* en todas partes y usar estructuras de datos inmutables. En Vue, las dependencias de un componente se rastrean automáticamente durante su procesamiento, por lo que el sistema sabe con precisión cuál o cuáles realmente necesitan volver a renderizarse.

Esto significa que las actualizaciones en Vue no optimizado serán mucho más rápidas que React sin optimizar.

En desarrollo:

Si bien el rendimiento en la producción es la medida más importante, ya que está directamente asociada con la experiencia del usuario final, la utilidad en el

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|-------------------------|----------------------|

proceso sigue siendo importante porque está asociado con la experiencia del desarrollador.

En la velocidad tanto Vue como React permanecen lo suficientemente rápidas en desarrollo para la mayoría de las aplicaciones normales. Sin embargo, al crear prototipos de animaciones o visualizaciones de datos a alta velocidad de cuadros, se observan casos de Vue manejando 10 fotogramas por segundo en desarrollo mientras React caía a aproximadamente 1 fotograma por segundo.

Angular 1

Parte de la sintaxis de Vue se verá muy similar a Angular (por ejemplo, `v-if` vs `ng-if`).

Se menciona una comparativa entre ambos:

- Vue es mucho más simple que Angular 1, tanto en términos de API como de diseño.
- Vue presenta una solución modular más flexible a la rigidez de Angular 1. Esto se lo puede observar como una ventaja o una desventaja. Sin embargo, Vue ofrece una plantilla de Webpack que podemos utilizar para mitigar esta desventaja.
- Angular 1 usa enlace bidireccional entre ámbitos, mientras que Vue impone un flujo de datos unidireccional entre los componentes. Esto permite el flujo de datos sea más fácil de razonar en aplicaciones no triviales.
- Vue usa un sistema de observación de rastreo de dependencias transparente con colas asincrónicas: todos los cambios se desencadenan independientemente a menos que tengan relaciones de dependencia explícitas. Angular utiliza un concepto llamado 'digest cycle', chequea si hay un cambio en todas las variables observadas por todos los `$scope` al modificar una variable, una especie de *loop*. La desventaja en Angular es que si hay más variables *watched* en una sola página, se produce un retardo al cargar la misma.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

Angular 2

Angular 2 presenta muchos cambios en comparación con Angular 1: el sistema de componentes, muchos detalles de implementación han sido completamente reescritos, y la API también fue completamente cambiada.

Mecanografiado:

Si bien Angular 1 podría usarse para aplicaciones más pequeñas, Angular 2 cambió de enfoque y está destinado a desarrollar las aplicaciones de grandes empresas. Como parte de esto, casi requiere TypeScript, otorgando un tipo de seguridad de lenguajes como Java y C#.

Para entornos empresariales Vue permite, de manera opcional, utilizar TypeScript a través de typings oficiales y decoradores oficiales. Hay poca documentación de la utilización de Vue en este entorno.

Tamaño y rendimiento:

En términos de rendimiento, ambos frameworks son rápidos. En cuanto al tamaño, aunque Angular 2 con compilación fuera de línea y *tree-shaking* es capaz de reducir considerablemente su tamaño, un Vue 2.0 con todas las funciones con compilador incluido (23kb) es aún más ligero que un ejemplo de Angular 2 deshilachado en un árbol (50kb). El tamaño de la aplicación Angular 2 es pequeño debido al *tree-shaking*, que elimina el código de las características que no está utilizando. Con el tiempo, volverá a crecer a su tamaño real a medida que importe y use más características del framework.

Curva de aprendizaje:

Para comenzar con Vue, todo lo que se necesita es familiarizarse con HTML y ES5 JavaScript (es decir, JavaScript simple). Con estas habilidades básicas, se puede comenzar a crear aplicaciones no triviales en menos de un día de lectura de la guía.

La curva de aprendizaje de Angular 2 es mucho más pronunciada. Incluso sin TypeScript, su guía de inicio rápido comienza con una aplicación que utiliza ES2015

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

JavaScript, NPM con 18 dependencias, 4 archivos y más de 3.000 palabras para explicarlo todo, solo para decir Hello World.

Ember

Ember es un framework completo que está diseñado para ser muy dogmático. Proporciona muchas convenciones establecidas y permite obtener una gran productividad. Sin embargo, la curva de aprendizaje es alta y, por lo tanto, la flexibilidad sufre.

Se realiza a continuación una comparativa entre el núcleo de Vue y las capas de modelado de objetos y plantillas de Ember:

- Vue proporciona una discreta reactividad en objetos JavaScript simples y propiedades computadas completamente automáticas. En Ember, debe envolver todo en Ember Objects y declarar manualmente dependencias para las propiedades calculadas.
- La sintaxis de la plantilla de Vue aprovecha toda la potencia de las expresiones de JavaScript, mientras que en comparación, la expresión de la barra de herramientas y la sintaxis del ayudante es intencionadamente bastante limitada.
- En cuanto al rendimiento, incluso después de la última actualización del motor Glimmer en Ember 2.0., Vue carga automáticamente las actualizaciones, mientras que Ember necesita administrar manualmente los bucles de ejecución en situaciones de rendimiento crítico.

Polymer

Polymer es otro proyecto patrocinado por Google. Los componentes de Vue y los elementos personalizados de Polymer ofrecen un estilo de desarrollo muy similar. La mayor diferencia es que Polymer se basa en las últimas características de los componentes web y requiere que los polyfills no triviales funcionen (con un

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

rendimiento degradado) en navegadores que no admiten esas características de forma nativa. Por el contrario, Vue funciona sin dependencias ni rellenos hasta IE9.

En Polymer 1.0, el sistema de enlace de datos es muy limitado para compensar el rendimiento. Por ejemplo, las únicas expresiones admitidas en las plantillas de Polymer son la negación booleana y las llamadas a métodos únicos.

Los elementos personalizados de Polymer se crean en archivos HTML, lo que lo limita a JavaScript/CSS simple (y características de idioma admitidas por los navegadores actuales). Los componentes de un solo archivo de Vue permiten usar ES2015 + y cualquier preprocesador de CSS.

Riot

Riot 2.0 proporciona un modelo de desarrollo basado en componentes similares (que se llama "tag") con una API mínima.

A pesar de ser un poco más pesado que Riot, Vue ofrece algunas ventajas:

- Sistema de efecto de transición. Riot no tiene ninguno.
- Un enrutador más potente. La API de enrutamiento de Riot es mínima.
- Riot atraviesa un árbol DOM en lugar de usar un DOM virtual, por lo que sufre los mismos problemas de rendimiento que Angular 1.
- Vue proporciona soporte oficial para Webpack y Browserify, mientras que Riot solo se apoya en el soporte de la comunidad para la integración del sistema de compilación.

3.1.2. Elección de Vue.js

Con la comparativa anterior podemos justificar la elección de Vue JS para la realización de nuestra aplicación web progresiva, demostrando que dentro de la oferta de distintos frameworks es el que más se adapta a nuestra necesidad.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

3.2. Firebase

Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada por Firebase, Inc. en 2011, y adquirida por Google en 2014.

Presta diferentes servicios en el área de *Analytics, Develop, Grow y Earn*.

Para los fines del presente trabajo se va a hacer uso de tres servicios de Firebase: *Realtime Database, Firebase Storage y Firebase Auth*.

Firebase permite:

- Compilar aplicaciones más rápido, sin administrar la infraestructura. No hay necesidad de administrar servidores.
- Contar con la infraestructura de Google que se escala automáticamente con la aplicación.
- Tener una base de datos en tiempo real.
- Alojamiento de información en Google Cloud Storage directamente desde el cliente.
- Autenticación de clientes rápidamente. Permite OAuth2 para Google, Facebook, Twitter y GitHub.

3.2.1. Ventajas y Desventajas

Ventajas

- Autenticación mediante Email y contraseña, Google, Facebook, o Github.
- Base de datos en tiempo real.
- API confeccionada.
- Seguridad integrada en el nivel del nodo de datos.
- Almacenamiento de archivos respaldado por Google Cloud Storage.
- Alojamiento de archivos estáticos.
- Tratar los datos como flujos para crear aplicaciones altamente escalables.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

- No hay que preocuparse por la infraestructura.

Desventajas

- Habilidades de consulta limitadas debido al modelo de flujo de datos de Firebase.
- Los modelos tradicionales de datos relacionales no son aplicables a NoSQL.
- Sin instalación en el lugar.

3.2.2. Realtime Database

Firestore proporciona una base de datos en tiempo real y back-end como un servicio. Es una base de datos NoSQL alojada en la nube que permite almacenar y sincronizar datos entre los usuarios en tiempo real. El servicio proporciona a los desarrolladores de aplicaciones una API que permite que los datos de las aplicaciones se sincronicen entre los clientes y se almacenen en la nube de Firestore. La compañía proporciona bibliotecas de clientes que permiten la integración con aplicaciones Android, iOS, JavaScript, Java, Objective-C, swift y Node.js. También se puede acceder a la base de datos a través de una API REST y enlaces para varios frameworks de JavaScript como AngularJS, React, Ember.js, Backbone.js y Vue.js. La API REST utiliza el protocolo de eventos enviados por el servidor, que es una API para crear conexiones HTTP para recibir notificaciones push de un servidor. Los desarrolladores que usan la base de datos en tiempo real pueden proteger sus datos utilizando las reglas de seguridad implementadas en el lado del servidor de la compañía.

Firestore se destaca, y se diferencia, por contar con las siguientes características:

- Permite colaborar entre dispositivos con facilidad: La sincronización en tiempo real permite que los usuarios accedan a los datos desde cualquier dispositivo, web o móvil, y los ayuda a trabajar en conjunto.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

- Optimizado para el uso sin conexión: Cuando los usuarios se desconectan, los SDK de Realtime Database usan la caché local del dispositivo para publicar y almacenar cambios. Cuando el dispositivo se conecta, los datos locales se sincronizan de manera automática.
- Posee seguridad sólida basada en usuarios: Realtime Database se integra con Firebase Authentication para brindar autenticación intuitiva y sencilla para los programadores. Se puede usar el modelo de seguridad declarativo para permitir el acceso según la identidad de los usuarios o con patrones que coinciden con los datos.

3.2.3. Firebase Storage

Firebase Storage proporciona cargas y descargas seguras de archivos para las aplicaciones de Firebase, independientemente de la calidad de la red. El desarrollador puede usarlo para almacenar imágenes, audio, video u otro contenido generado por el usuario. Firebase Storage está respaldado por Google Cloud Storage.

Firebase permite al desarrollador:

- Crear a la misma escala que Google: Contiene una infraestructura diseñada para el caso en que la aplicación se vuelva viral.
- Cargas y descargas robustas: el SDK de Firebase para Cloud Storage permite detener y reanudar las transferencias de forma automática cuando la app pierda y recupere la conectividad móvil, lo cual ahorra tiempo y ancho de banda a los usuarios.
- Seguridad sólida basada en usuarios: El SDK de Firebase para Cloud Storage se integra con Firebase Authentication para proporcionar un control de acceso.

La elección de Firebase viene ligada a la necesidad de FluxIT de sumar nuevas tecnologías a su oferta de soluciones. Como se detalló anteriormente, Firebase

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

presenta servicios de *cloud-computing*, es decir, un paradigma en auge que libera la dependencia de un servidor *on site* y aloja los datos en servidores ajenos.

3.2.4. Firebase Auth

Firebase Auth es un servicio que puede autenticar usuarios que usan solo el código del lado del cliente. Es compatible con los proveedores de inicio de sesión social Facebook, GitHub, Twitter y Google. También permite el inicio de sesión mediante números de teléfono. Además, incluye un sistema de administración de usuarios mediante el cual los desarrolladores pueden habilitar la autenticación del usuario con el inicio de sesión de correo electrónico y contraseña almacenado con Firebase.

Firebase Authentication se integra estrechamente con otros servicios de Firebase y aprovecha los estándares de la industria como OAuth 2.0 y OpenID Connect.

4. Diseño, desarrollo e implementación de un sistema aplicado al control de beneficios sociales en FluxIT

El sistema finalizado debe posibilitar la realización de los siguientes ítems:

- Observar la cantidad de *recoins* y su respectivo vencimiento.
- Obtener un detalle de cada movimiento de *recoins*.
- Gestionar el pedido de premios.
- Observar el estado del premio canjeado.
- Modificar y autorizar canjes.
- Observar las tareas que brindan *recoins* y un detalle de las mismas.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

- Enviar recoins a partir de *feedbacks* y tareas.
- Gestionar nuevos premios y nuevas tareas.
- Enviar notificaciones push y toastr ante distintos eventos.
- Enviar correos corporativos por eventos importantes (a definir).

La aplicación está desarrollada con tecnologías web y se destaca por la implementación de Vue.js. Otra librería de gran importancia es Vuefire la cual permite conectar la aplicación web al servicio Firebase.

La misma está implementada en una base de datos NoSQL alojada en Firebase. Contiene documentos que están dentro de colecciones cuyo nombre es una *key* única que define Firebase, estas a su vez están dentro de otras colecciones que pueden definirse. Se observa un ejemplo de esto en la imagen 2.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|



Imagen 2: Formato base de datos en Firebase

El sistema posee conexiones web que permiten consultar y cargar nuevos datos. Posee dos clases de usuario:

1. Usuario general: Lo poseen todos los *Fluxers*. Permite la consulta de movimientos y canjes, gestionar sus premios y visualizar las tareas que brinden *recoins*. También permite enviar *recoins* a otro *Fluxers* mediante un *feedback*. La cantidad de *recoins* disponibles es limitada.
2. Usuario administrador: Lo posee el personal de Valor Humano y permite la administración (alta, baja, modificación) de premios, tareas y canjes. Puede brindar *recoins* a los *Fluxers* a través de las tareas.

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|-------------------------|----------------------|

El proyecto se dividió en sprints⁵ (iteraciones). Se estimó una cantidad de 5 o 6 sprints de una duración aproximada de dos semanas. Cada sprint resulta en un entregable que es presentado a los *stakeholders*⁶ al finalizar el mismo.

Se describe los objetivos principales de cada sprint:

1. Realizar el diseño del sistema y definir historias de usuario generales. Comenzar la codificación de la funcionalidad básica del usuario general.
2. Codificar la funcionalidad básica del usuario administrador.
3. Concluir la funcionalidad básica del circuito; Los fluxers pueden brindarse recoins entre ellos, gestionar el pedido de premios y el administrador puede finalizar dichos pedidos.
4. Permitir carga de imágenes en nuevas tareas y premios.
El usuario administrador puede enviar recoins a partir de una tarea y el usuario general ahora tiene una cantidad limitada de recoins para enviar.
5. Crear las funciones para recargar los recoins disponibles que pueden enviarse los Fluxers y para verificar el vencimiento de los recoins personales. También se agregan notificaciones toast y push.
6. Refactorizar el código y realizar ultimas pruebas de funcionalidad.

A su vez, en cada sprint se realizan modificaciones a la base de datos y al sistema de archivos.

Las historias de usuario son definidas en el primer sprint. También se añaden algunas luego de finalizar el sprint y presentar la funcionalidad a los stakeholders, los mismos añaden comentarios y futuros cambios a la aplicación.

⁵ El Sprint es el período en el cual se lleva a cabo el trabajo en sí. Es recomendado que la duración de los sprints sea constante y definida por el equipo con base en su propia experiencia. Al final de cada sprint, el equipo deberá presentar los avances logrados, y el resultado obtenido es un producto que, potencialmente, se puede entregar al cliente.

⁶ Stakeholders (Clientes, Proveedores, Vendedores, etc): Son las personas que hacen posible el proyecto y para quienes el proyecto producirá el beneficio acordado que justifica su desarrollo. Sólo participan directamente durante las revisiones del "sprint".

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

4.1. 1º Sprint

Se añaden las historias de usuario generales al *Trello* y se define el alcance general del proyecto.

Como referencia, en la imagen 3 se observa el dashboard de Trello previo a finalizar un sprint. En la misma se detectan varias columnas:

- Backlog: Historias a desarrollar en futuros sprints.
- Planificado Sprint: Las historias que vamos a realizar en el presente sprint.
- En desarrollo: La historia que estamos desarrollando en este momento.
- Finalizado Sprint: Historias finalizadas en el sprint.
- DONE: Historias archivadas finalizadas en sprints anteriores.

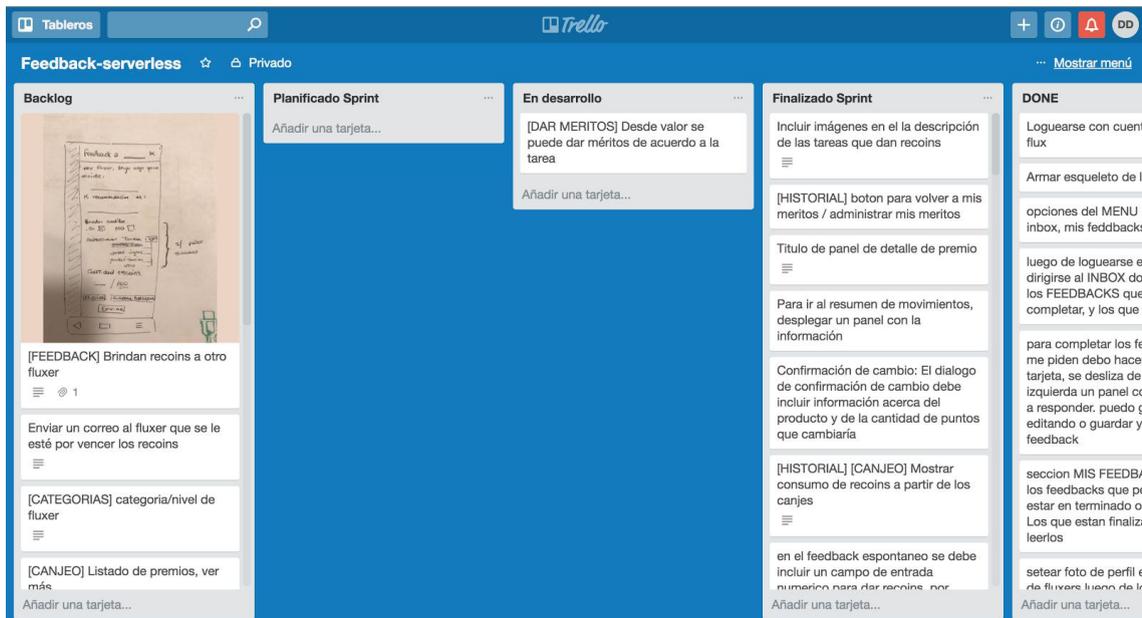


Imagen 3: Dashboard del Trello

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

También se desarrolló la funcionalidad básica del usuario general.

A la mitad del sprint se realizó una muestra a Valor Humano para poder definir los componentes visuales de la aplicación. Se observa en la imagen 4 el boceto presentado. Durante esta muestra la aplicación no está conectada a la base de datos, solo se presentó un borrador y el formato de tarjetas que podría tener. A partir de esta muestra y de la devolución de Valor Humano, se confirmó dicho formato.

El sprint concluyó con la conexión de la aplicación a la base de datos, se completó su funcionalidad y se finalizó el formato de tarjetas para presentar la información. En la imagen 5 se observa el dashboard del usuario general en formato desktop y en la imagen 6 en formato mobile.

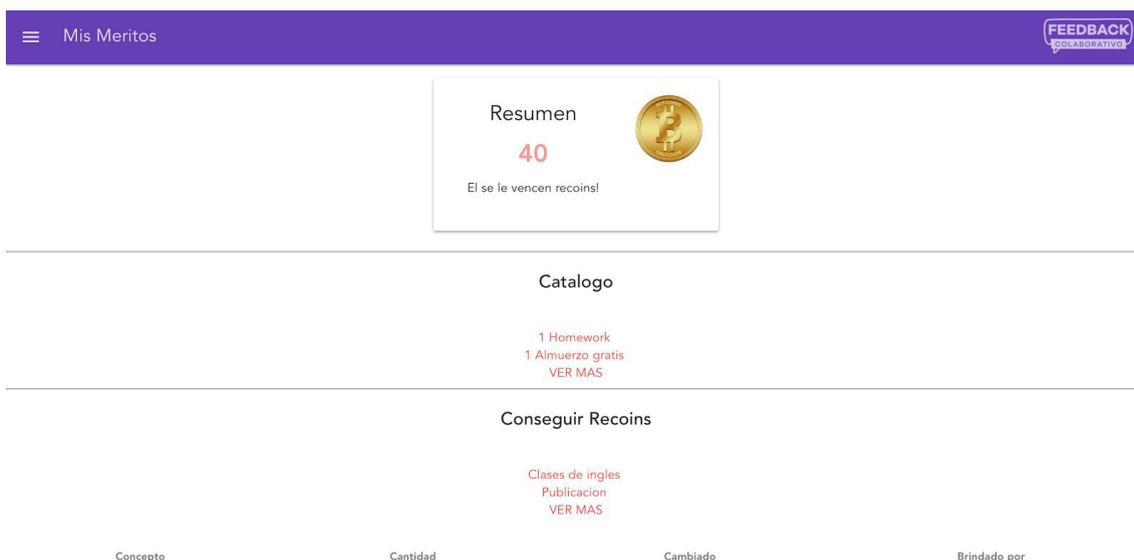


Imagen 4: Boceto del dashboard del usuario general.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

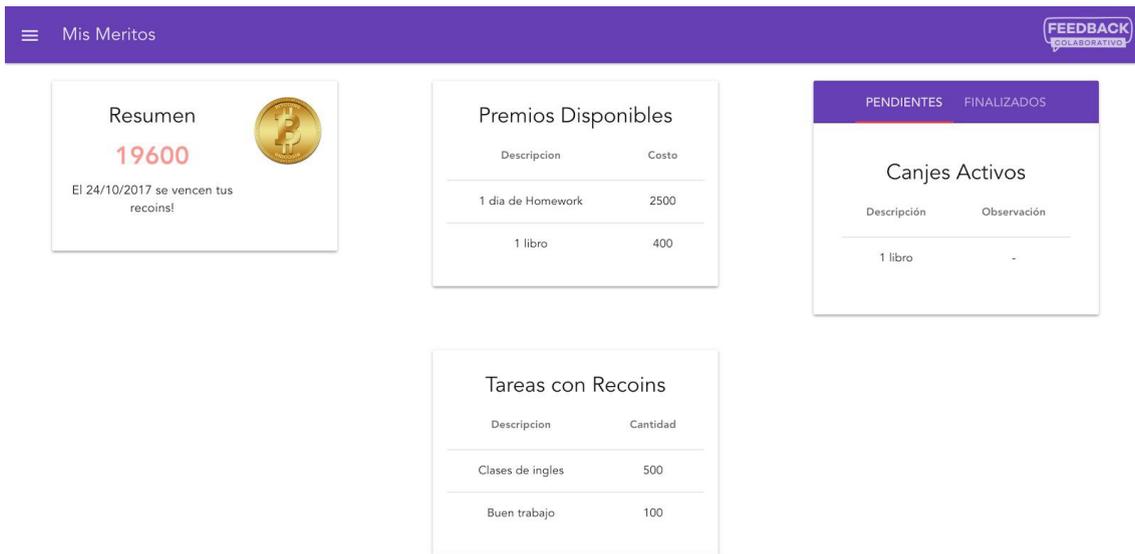


Imagen 5: Dashboard del usuario general (desktop).



Imagen 6: Dashboard del usuario general (mobile).

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|-------------------------|----------------------|

Como se observa, al finalizar el primer sprint, el usuario puede ver un resumen de su vencimiento (aún no funcional) y de la cantidad de recoins que tiene. También permite advertir los premios disponibles, las tareas con recoins y sus canjes activos y finalizados. A excepción del vencimiento, el resto de datos son consultados a la base de datos.

El usuario también puede acceder a ver el detalle de las tareas que brindan los recoins y al detalle de los premios, que le permite realizar el canje. Se puede observar en la imagen 7 y 8.

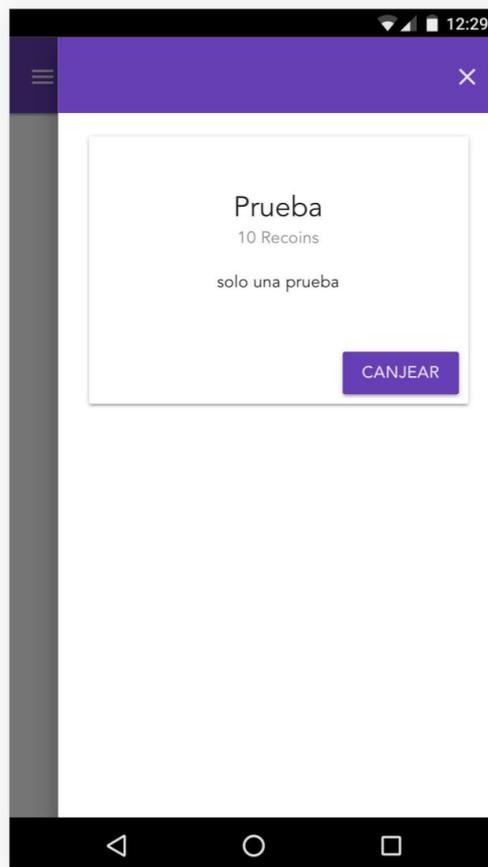


Imagen 7: Detalle premio.

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |

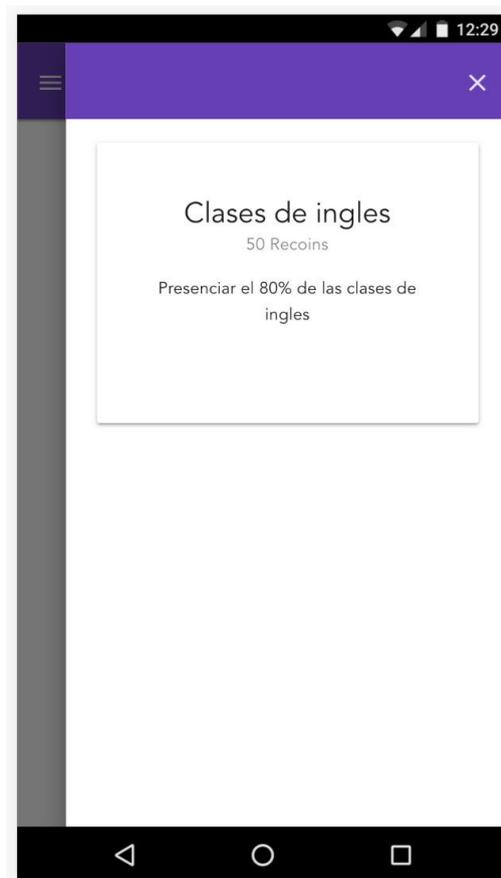


Imagen 8: Detalle tarea.

4.2. 2º Sprint

En el segundo sprint se desarrollaron las funcionalidades básicas del usuario administrador. Se creó este usuario nuevo que, al finalizar el presente sprint, permitió realizar las siguientes actividades:

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |

- Autorizar el canje de premios. También permite agregar observaciones (imagen 9).
- Modificar y borrar premios y tareas (imagen 10).
- Dar de alta premios y tareas (imagen 11).

También se desarrolló el dashboard del usuario. Se lo puede observar en la imagen 12 y 13.



Imagen 9: Detalle Canje.

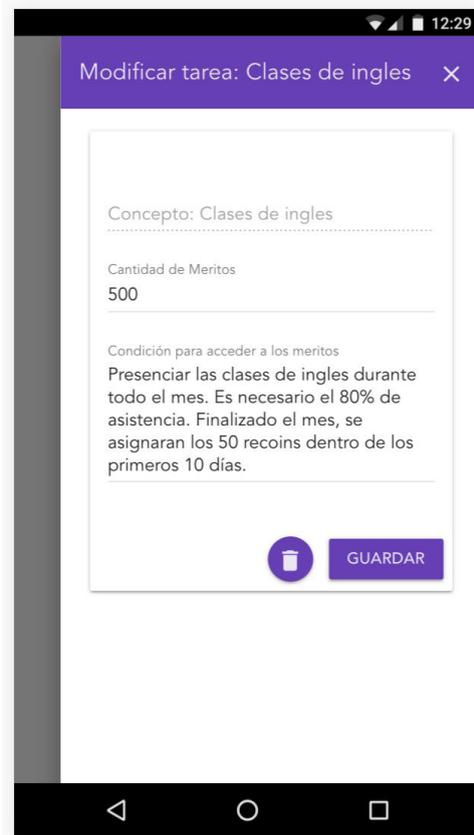
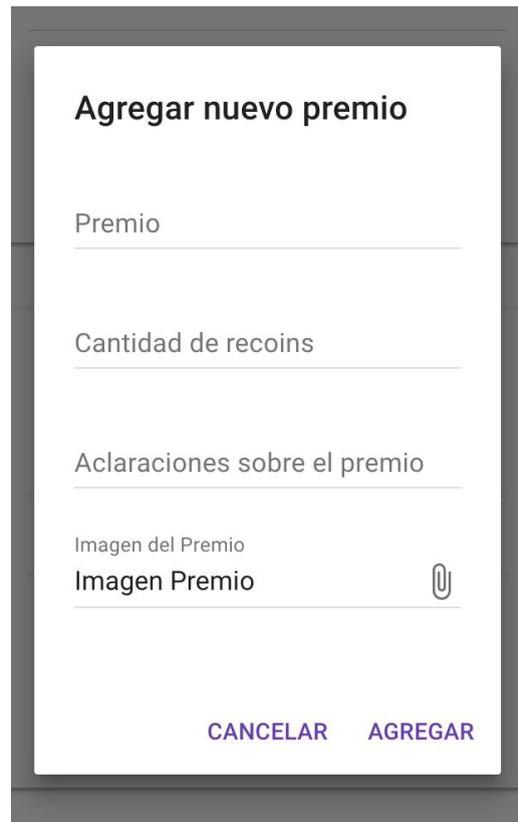


Imagen 10: Detalle tarea - Administrador.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |



Agregar nuevo premio

Premio

Cantidad de recoins

Aclaraciones sobre el premio

Imagen del Premio

Imagen Premio 

CANCELAR **AGREGAR**

Imagen 11: Crear Premio.

Si bien existe un campo para adjuntar una imagen del premio, o tarea, está funcionalidad junto con la de validación de formulario no están desarrolladas en esta instancia.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |

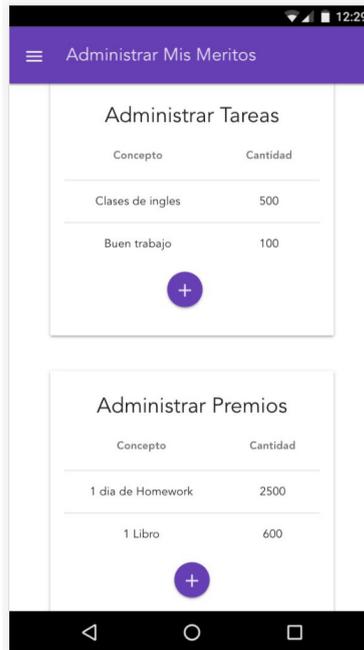


Imagen 12: Dashboard Administrador (mobile).

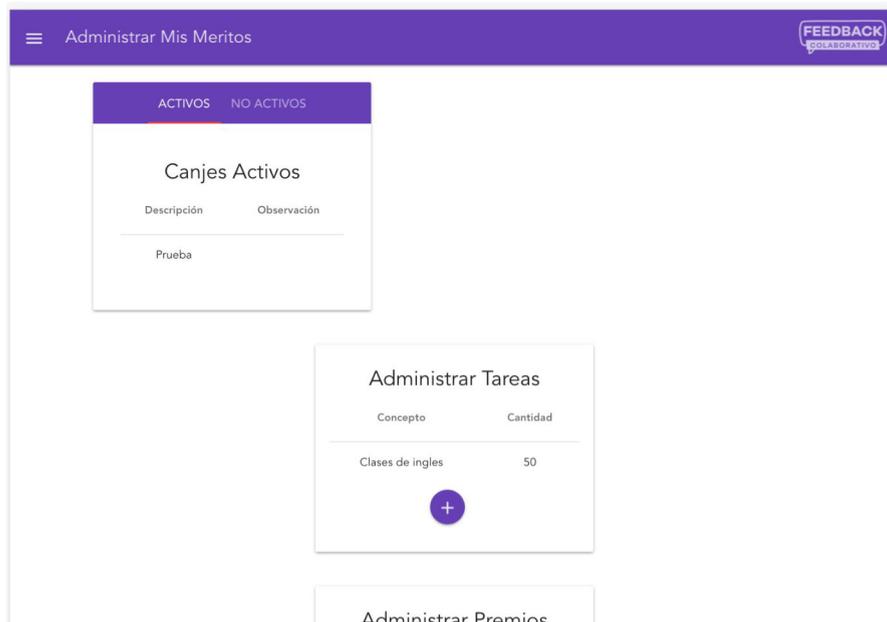


Imagen 13: Dashboard Administrador (desktop).

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |

Al finalizar este sprint, se observaron los siguientes errores:

- El dashboard del usuario administrador, en la versión desktop, presenta errores visuales.
- La carga de nuevos premios y tareas todavía no permite adjuntar imágenes.
- Las modificaciones que sufren cambios pero no son guardados, mantienen el estado modificado hasta actualizar la página.

4.3. 3º Sprint

Durante el tercer sprint, el objetivo principal fue concluir el ciclo de realimentación. Hasta el segundo sprint, los recoins con los que contaban los Fluxers para gastar eran asignados directamente desde la base de datos. A partir de esto, los Fluxers pueden enviarse recoins a través de feedbacks espontáneos (imagen 14 y 15). Por el momento, la cantidad de recoins que el usuario puede enviar es ilimitada.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

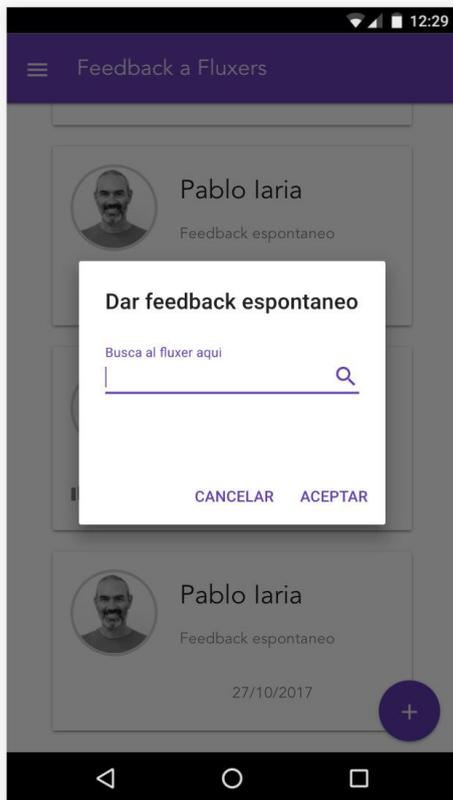


Imagen 14: Inicio Feedback espontáneo.

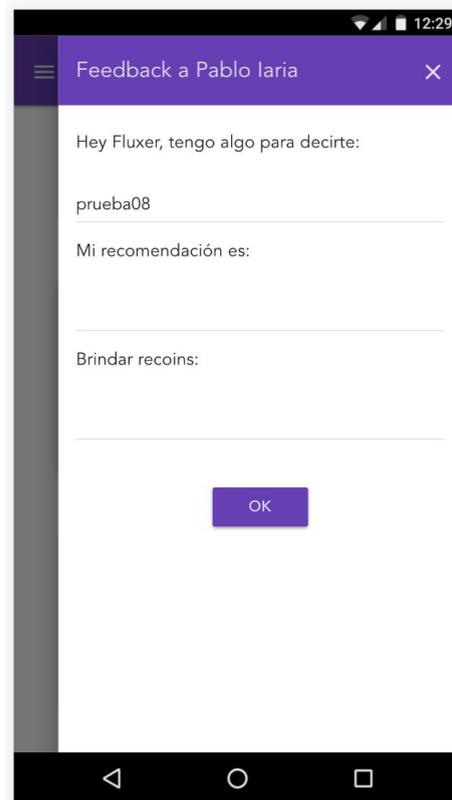


Imagen 15: Detalle Feedback espontáneo.

Durante este sprint también se arreglaron los detalles surgidos en sprints anteriores. Al finalizar, se hizo una pequeña refactorización del código, garantizando de esta manera un óptimo mantenimiento del mismo.

4.4. 4º Sprint

En este sprint, se permite cargar imágenes a nuevos premios y tareas. Estas imágenes son subidas al servicio *Storage* de Firebase con el nombre de la *key*, del

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |

premio o la tarea, alojada en el servicio *Database*. De esta manera, a pesar de ser servicios distintos, al cargar un premio o tarea también se realiza la carga de dicha imagen sin ningún conflicto. En la imagen 16 se muestra la carpeta de imágenes de premios en Firebase Storage y en la imagen 17, un ejemplo para subir una imagen.

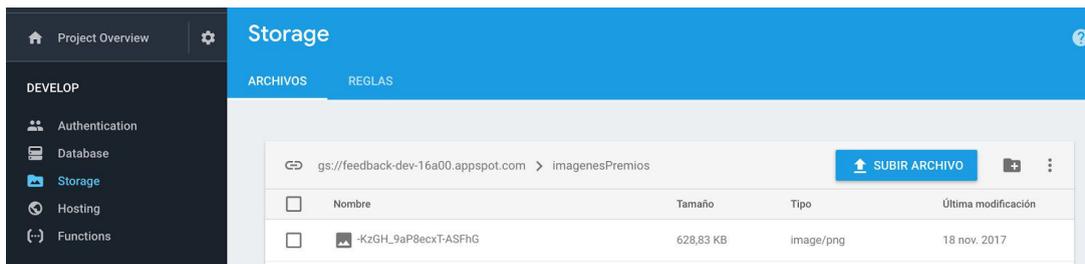


Imagen 16: Firebase Storage

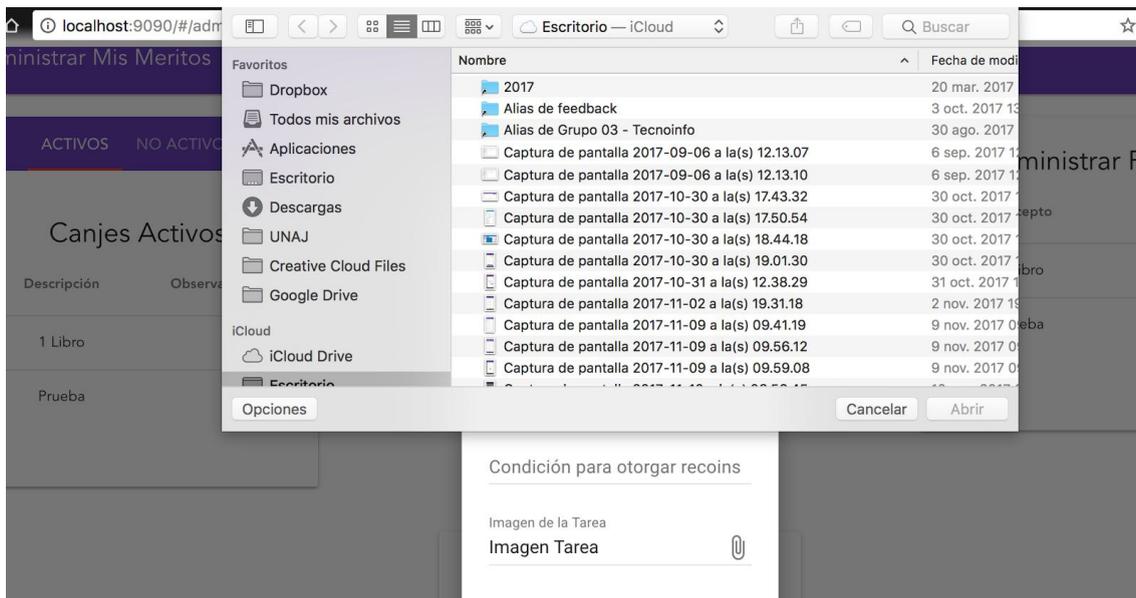


Imagen 17: Adjuntar imagen

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |

En las imágenes 18, 19 y 20 se puede observar el proceso para enviar recoins a los fluxers a partir de una tarea completada. Con esta actividad, Valor Humano puede brindar recoins a quien considere que haya finalizado un trabajo.



Imagen 18: Detalle de tarea

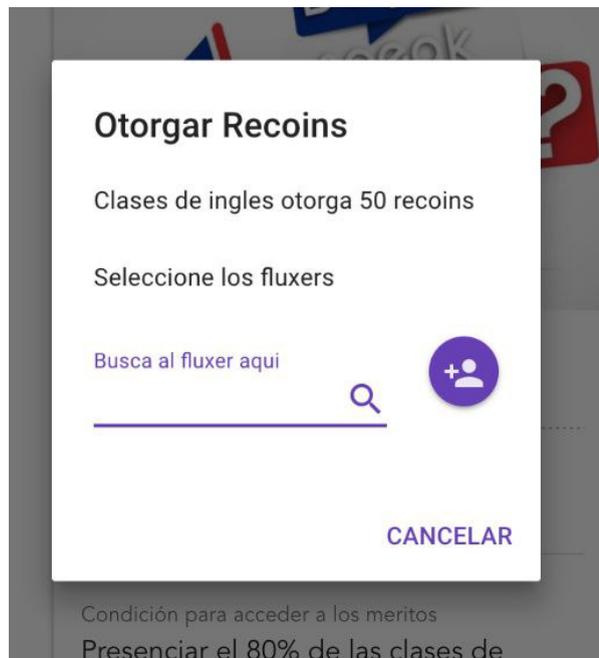


Imagen 19: Seleccionar Fluxer (1)

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |

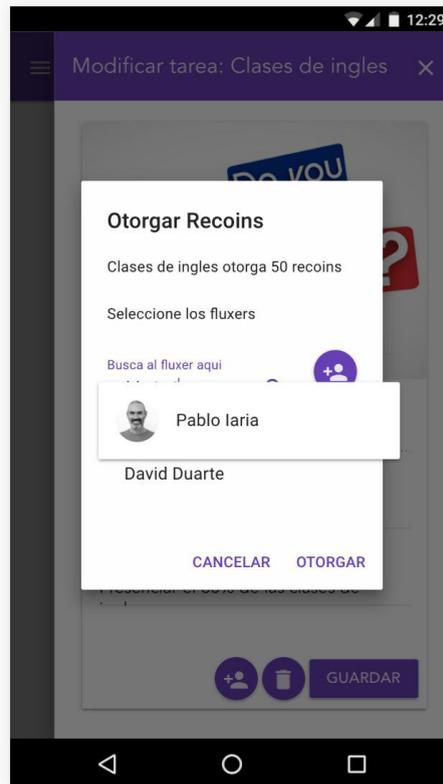


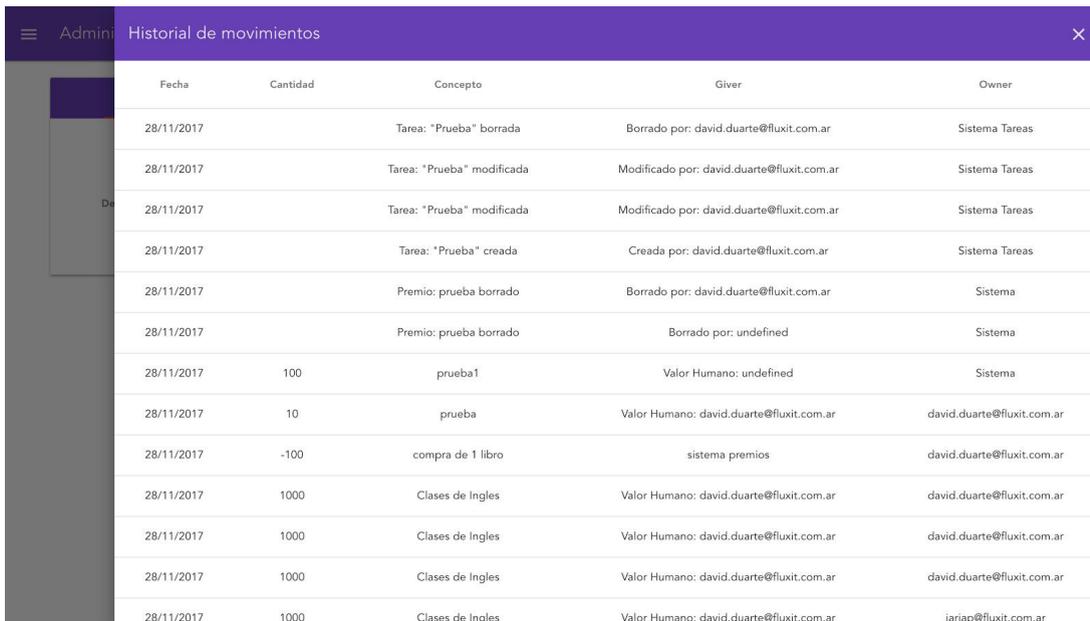
imagen 20: Seleccionar Fluxer (2)

Otra funcionalidad que se sumó fue el historial de movimientos. El usuario administrador puede consultar todos los producidos en el sistema: desde las altas bajas y modificaciones de premios, tareas y canjes; hasta la asignación de recoins por tareas o por feedbacks.

En cuanto al usuario general, éste puede consultar solamente los movimientos personales: recoins asignados, modificaciones a premios canjeados y autorización o modificación de los mismos.

En la imagen 21 se observa el historial de movimientos del usuario administrador.

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |



| Fecha | Cantidad | Concepto | Giver | Owner |
|------------|----------|----------------------------|--|----------------------------|
| 28/11/2017 | | Tarea: "Prueba" borrada | Borrado por: david.duarte@fluxit.com.ar | Sistema Tareas |
| 28/11/2017 | | Tarea: "Prueba" modificada | Modificado por: david.duarte@fluxit.com.ar | Sistema Tareas |
| 28/11/2017 | | Tarea: "Prueba" modificada | Modificado por: david.duarte@fluxit.com.ar | Sistema Tareas |
| 28/11/2017 | | Tarea: "Prueba" creada | Creada por: david.duarte@fluxit.com.ar | Sistema Tareas |
| 28/11/2017 | | Premio: prueba borrado | Borrado por: david.duarte@fluxit.com.ar | Sistema |
| 28/11/2017 | | Premio: prueba borrado | Borrado por: undefined | Sistema |
| 28/11/2017 | 100 | prueba1 | Valor Humano: undefined | Sistema |
| 28/11/2017 | 10 | prueba | Valor Humano: david.duarte@fluxit.com.ar | david.duarte@fluxit.com.ar |
| 28/11/2017 | -100 | compra de 1 libro | sistema premios | david.duarte@fluxit.com.ar |
| 28/11/2017 | 1000 | Clases de Ingles | Valor Humano: david.duarte@fluxit.com.ar | david.duarte@fluxit.com.ar |
| 28/11/2017 | 1000 | Clases de Ingles | Valor Humano: david.duarte@fluxit.com.ar | david.duarte@fluxit.com.ar |
| 28/11/2017 | 1000 | Clases de Ingles | Valor Humano: david.duarte@fluxit.com.ar | david.duarte@fluxit.com.ar |
| 28/11/2017 | 1000 | Clases de Ingles | Valor Humano: david.duarte@fluxit.com.ar | iariap@fluxit.com.ar |

Imagen 21: Historial movimientos.
Usuario administrador.

4.5. 5º Sprint

En este sprint, se suman las notificaciones toast a todos los eventos que ocurran tanto de alta, baja o cualquier modificación.

En las imágenes 22, 23 y 24 se puede observar algunos ejemplos de notificaciones Toast.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|



Imagen 22: Toast al brindar puntos por tarea

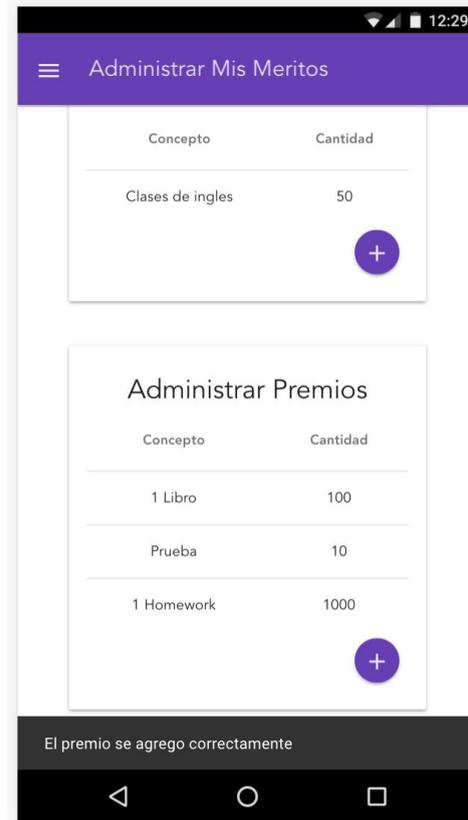


Imagen 23: Toast al agregar premio

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |

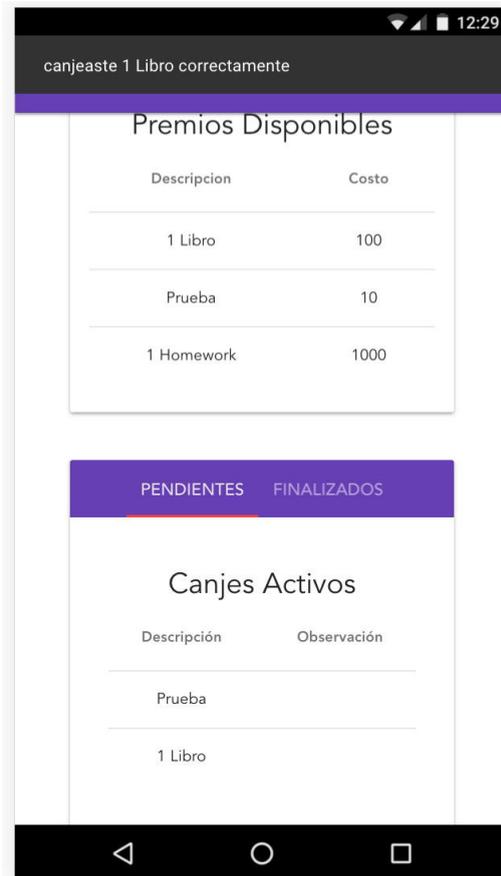


Imagen 24: Toast al canjear premio

Las notificaciones Push no fueron agregadas debido a que se necesita un acceso por vía https. Hasta este sprint se sigue trabajando en un entorno de desarrollo local (localhost) que se accede por http.

Aun así, como la aplicación de Feedback ya tiene implementada estas notificaciones y la misma está en un entorno funcional, implementarlas no son de gran complejidad.

Además, se creó una función para recargar los recoins disponibles que pueden enviarse los Fluxers y otra para las notificaciones por e-mail. La primera se va a

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|-------------------------|----------------------|

ejecutar cada cierto tiempo a determinar por Valor Humano, y respecto a la segunda resta definir, con el mismo sector, en qué momentos se va a utilizar. Hay que recordar que los dispositivos iOS no pueden recibir notificaciones push y es necesario algún método alternativo para notificar a estos usuarios de eventos importantes, como el vencimiento de los recoins. Se muestran dichas funciones en la imagen 25.

Para el caso de la función de envío de correos, cuando se implemente se debe pasar el mismo por parámetro. En la imagen se observa que es enviado a uno en particular, utilizado para confirmar que la función esté desarrollada correctamente.

Se comprobó el funcionamiento de estas ejecutando una consulta *curl* a través de la consola de NodeJS. Ver imagen 26 y 27.

```
app.post('/recargar', (req, resp) => {
  fluxers.orderByChild("email").on("child_added", function(s, prevChild) {
    if (s.val().lider) {
      fluxers.child(s.key + '/puntosDisponibles').set(100)
    } else {
      fluxers.child(s.key + '/puntosDisponibles').set(55)
    }
  })
})

app.post('/enviar', (req, resp) => {
  sendEmail('david.duarte@fluxit.com.ar')
})

function sendEmail(email) {
  const mailOptions = {
    from: 'no-reply@fluxit.com.ar', // sender address
    to: email, // list of receivers
    subject: 'Actualización en Feedback', // Subject line
    html: 'Tiene una nueva actualización en el sistema de Meritos. Ingrese a la brevedad//';
  };
  return transporter.sendMail(mailOptions, function (err, info) {
    if(err)
      console.log(err)
    else
      console.log(info);
  });
}
```

Imagen 25: Funciones agregadas

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |

```

[MacBook-Pro-de-mbp:feedback David$ firebase deploy --only functions
]

=== Deploying to 'feedback-dev-16a00'...

i  deploying functions
i  functions: ensuring necessary APIs are enabled...
i  runtimeconfig: ensuring necessary APIs are enabled...
✓  runtimeconfig: all necessary APIs are enabled
✓  functions: all necessary APIs are enabled
i  functions: preparing functions directory for uploading...
i  functions: packaged functions (2.21 KB) for uploading
✓  functions: functions folder uploaded successfully
i  starting release process (may take several minutes)...
i  functions: updating function api...
✓  functions[api]: Successful update operation.
✓  functions: all functions deployed successfully!

✓  Deploy complete!

Project Console: https://console.firebase.google.com/project/feedback-dev-16a00/overview
Function URL (api): https://us-central1-feedback-dev-16a00.cloudfunctions.net/api
i

```

Imagen 26: Deploy del proyecto (sólo funciones)

```

[MacBook-Pro-de-mbp:feedback David$ curl -X POST -H "Content-Type:application/json" -d '' https://us-central1-feedback-dev-16a00.cloudfunctions.net/api/enviar

```

Imagen 27: Ejecución de consulta desde consola

4.6. 6º Sprint

Durante este sprint, se refactorizó el código, con el objetivo de lograr un mantenimiento más eficaz y cambios más rápidos.

Previo a lanzar una versión de producción, se subió el proyecto a un ambiente de testing y se realizaron las últimas pruebas con sus respectivos arreglos.

Finalizadas las mismas y completado los últimos arreglos, se presentó una última demostración a los stakeholders. Con esto, se concluyó la ampliación a la aplicación de Feedback.

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |

5. Conclusión: problemas generales surgidos al realizar la aplicación

Trabajar con herramientas desconocidas previamente, y en un equipo nuevo, presentó diversas problemáticas: desde la decisión de la metodología a utilizar hasta la propia codificación.

Se listan algunas de las problemáticas “ajenas” a la programación:

- Diagramar la táctica del trabajo fue el primer desafío. La empresa utiliza una metodología similar a Scrum, que es ágil y flexible diseñada para el desarrollo de software. Scrum, entre otras cosas plantea la conformación de equipos pequeños de entre 5 y 9 personas. El grupo encargado de desarrollar la aplicación estuvo conformado sólo por dos personas, por lo que se adaptó el procedimiento a las necesidades del mismo, evitando las *daily standup* y los *daily scrum*.
- Planificar las iteraciones y sus tareas también formó parte de las primeras problemáticas. No contar con experiencia previa en las herramientas (Vue y Firebase) imposibilitó una correcta planificación de los primeros sprints. Para el primero se desarrollaron más del doble de tareas planificadas. Una rápida curva de aprendizaje que provee Vue y el hecho de que se comenzó a trabajar sobre una aplicación ya desarrollada, permitió un desarrollo veloz y entregables con muchas funcionalidades, incluso a partir del primer sprint.
- Recolección de información sobre los requerimientos de la aplicación y su funcionalidad. Para esto, se realizaron reuniones con personal de Valor Humano, ajenos a los conocimientos de programación. La recolección de información fue llevada a cabo por ambos integrantes del equipo, por lo tanto, debieron tomar el rol de analistas funcionales.

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|-------------------------|----------------------|

Comprender los requisitos de Valor Humano y trasladarlos a historias de usuarios formó parte de uno de los desafíos de cada sprint.

Problemas al momento de la programación:

- En el caso de las herramientas, previamente se destacó que Vue cuenta con una curva de aprendizaje muy rápida, esto permitió comenzar a desarrollar la aplicación a la brevedad. Aún así presentó el desafío de ser una herramienta no utilizada previamente.

En cuanto a Firebase, provee una documentación de la api muy clara pero aun así, adaptarla a una aplicación diseñada con Vue fue complejo. Para esto, se utilizó la librería Vuefire que facilitó la conexión de los servicios de Firebase a la aplicación.

Otro desafío que presentó Firebase Database, con las consultas a la base de datos, fue ordenar las tablas. Este servicio permite ordenarlas de acuerdo a una columna en particular pero no deja cambiar a un orden descendente; una consulta que en sql se resume en una sola línea de código (*order by ... desc*) en Firebase no está implementado. Para resolver esto se insertó en cada objeto una nueva clave denominada "orden", que consiste en tomar como valor un objeto *Date* en negativo. Ordenando a partir de esta columna, forzamos tener el último elemento insertado en el primer lugar.

- La decisión de realizar una extensión a una aplicación ya existente, realizada por otro desarrollador, presentó el desafío de comprender lo realizado y que se mantenga el mismo formato de codificación.

| | | | |
|-------------------|-------------------|----------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|----------------------------|----------------------|

6. Bibliografía utilizada

Alphabet Inc. (29 de noviembre de 2017). Progressive Web Apps. Recuperado de: <https://developers.google.com/web/progressive-web-apps/?hl=es>

Aulaformativa.com. (29 de noviembre de 2017). ¿Qué son las aplicaciones web progresivas y cuáles son sus características? Recuperado de: <http://blog.aulaformativa.com/que-son-aplicaciones-web-progresivas-caracteristicas/>

Álvarez Caules, Cecilio. (29 de Noviembre de 2017). ¿Qué es el Virtual DOM y como funciona?. Recuperado de: <https://www.arquitecturajava.com/que-es-el-virtual-dom-y-como-funciona/>

Esplin, Chris. (4 de Diciembre de 2017). What is Firebase. Recuperado de: <https://howtofirebase.com/what-is-firebase-fcb8614ba442>

Ghosh, Deepanjan. (4 de Diciembre de 2017). What is dirty checking in AngularJs and how does it work? Recuperado de: <https://www.quora.com/What-is-dirty-checking-in-AngularJs-and-how-does-it-work>

You, Evan (2014-2017). (29 de Noviembre de 2017). The Progressive JavaScript Framework. Recuperado de: <https://vuejs.org/>

You, Evan (2014-2017). (29 de Noviembre de 2017). Introduction. Recuperado de: <https://vue-loader.vuejs.org/en/>

You, Evan (2014-2017). (29 de Noviembre de 2017). Introducción. Recuperado de: <https://es-vuejs.github.io/vuejs.org/v2/guide/>

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |

Esaú, Abril. (4 de Diciembre de 2017). SourceForge vs GitHub vs Bitbucket vs GitLab. Recuperado de: <https://openwebinars.net/blog/sourceforge-vs-github-vs-bitbucket-vs-gitlab/>

Firebase, Inc. (4 de Diciembre de 2017). Cloud Storage para Firebase. Recuperado de: <https://firebase.google.com/products/storage/?hl=es-419>

Firebase, Inc. (4 de Diciembre de 2017). Firebase Realtime Database. Recuperado de: <https://firebase.google.com/products/database/?hl=es-419>

Firebase, Inc. (4 de Diciembre de 2017). Firebase te ayuda a crear mejores apps para dispositivos móviles y hacer crecer tu empresa. Recuperado de: <https://firebase.google.com/?hl=es-419>

Schatz, Jacob. (29 de Noviembre de 2017). Why we chose Vue.js. Recuperado de: <https://about.gitlab.com/2016/10/20/why-we-chose-vue/>

Velasco, Jairo. (29 de Noviembre de 2017). ¿Qué son las Aplicaciones Web Progresivas?. Recuperado de: <https://www.vitalinnova.com/aplicaciones-web-progresivas/>

Alarcón, José Manuel (29 de noviembre de 2017). Qué son las Aplicaciones Web Progresivas “Progressive Web Apps”. Recuperado de: <https://www.campusmvp.es/recursos/post/Que-son-las-Aplicaciones-Web-Progresivas-o-Progressive-Web-Apps.aspx>

Estrada, Kendru. (4 de Diciembre de 2017). Por qué es Vue.js el nuevo framework de moda. Recuperado de: <https://medium.com/blog-apside/por-que-es-vue-js-es-el-nuevo-framework-de-moda-79de70e13ef5>

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|-------------------------|----------------------|

Elizondo, Luis. Why we moved from Angular 2 to Vue.js (and why we didn't choose React). Recuperado de: <https://medium.com/reverdev/why-we-moved-from-angular-2-to-vue-js-and-why-we-didnt-choose-react-ef807d9f4163>

Bellucci, Marcelo. (4 de Diciembre de 2017). Se triplicó en cuatro años la cantidad de argentinos que tiene un celular inteligente. Recuperado de: https://www.clarin.com/sociedad/triplico-anos-cantidad-argentinos-celular-inteligente_0_S1638nVox.html

Moura, Marcos. (4 de Diciembre de 2017). Vue Material. Recuperado de: <http://vuematerial.io/>

Drasner, Sarah. (29 de Noviembre de 2017). Production Progressive Web Apps with JavaScript Frameworks. Recuperado de: <https://css-tricks.com/production-progressive-web-apps-javascript-frameworks/>

Software Freedom Conservancy. (4 de Diciembre de 2017). Fundamentos de Git - Deshacer Cosas. Recuperado de: <https://git-scm.com/book/es/v2/Fundamentos-de-Git-Deshacer-Cosas>

Trazada Internet Marketing 2016. (4 de Diciembre de 2017). Cinco razones para invertir en aplicaciones móviles. Recuperado de: <http://trazada.com/cinco-razones-para-invertir-en-aplicaciones-moviles/>

TU-APP.NET © 2016. (29 de Noviembre de 2017). Preguntas frecuentes sobre las Progressive Web Apps (PWA). Recuperado de: <https://soporte.tu-app.net/portal/kb/articles/preguntas-frecuentes-sobre-las-progressive-web-apps-pwa>

Hernandez, Uriel. (29 de Noviembre de 2017). Progressive web apps. Recuperado de: <https://codigofacilito.com/articulos/progressive-apps>

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
|-------------------|-------------------|-------------------------|----------------------|

Bueno, Uziel. (29 de Noviembre de 2017). Introduccion a Vue.js. Recuperado de: <https://uxweb.co/2015/11/17/introduccion-a-vue-js/>

Kumar, Vibhu. (4 de Diciembre de 2017). Que es Firebase. Recuperado de: <https://www.quora.com/What-is-firebase>

Ramirez, Vicente. (29 de Noviembre de 2017). QUE SON LAS PROGRESSIVE WEB APPS: VENTAJAS DE USAR ESTE FORMATO PARA TU APLICACIÓN. Recuperado de: <https://marketing4ecommerce.net/que-son-las-progressive-web-apps-marcando-el-futuro-mobile/>

Wikimedia Foundation, Inc. (29 de Noviembre de 2017). Vue.js. Recuperado de: <https://en.wikipedia.org/wiki/Vue.js>

Willis Towers, Watson. (4 de Diciembre de 2017). ¿Cómo han cambiado los beneficios sociales para empleados en los últimos 20 años?. Recuperado de: <https://willisupdate.com/20-anos-de-beneficios-sociales-para-empleados/>

Albaladejo, Xavier. (4 de Diciembre de 2017). Qué es SCRUM. Recuperado de: <https://proyectosagiles.org/que-es-scrum/>

| | | | |
|-------------------|-------------------|-------------------------|----------------------|
| Firma Estudiante: | Firma tutor UNAJ: | Firma tutor TAPTA UNAJ: | Firma tutor Empresa: |
| | | | |