



UNIVERSITAT DE
BARCELONA

Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

**Facultat de Matemàtiques i Informàtica
Universitat de Barcelona**

APLICACIÓN WEB AFA ESTALELLA

Enrique Miralles Yataco

Director: Karim Lekadir
Realitzat a: Departament de
Matemàtiques i Informàtica
Barcelona, 10 de junio de 2022

Resumen

En este trabajo de fin de grado, se realizará una implementación de una aplicación web que permitirá a la Asociación de Familias de Alumnos potenciar la comunicación familia-escuela.

Como padre y usuario he observado la necesidad de un sitio web en donde poder consultar la información recibida de la Asociación de Familias a quien de ahora en adelante llamaremos "AFA". Las herramientas de información actuales que utiliza la AFA son un grupo de WhatsApp y un apartado dentro de una web propiedad de la escuela que nos es bidireccional la cual se actualiza con poca frecuencia, como bien se sabe los grupos de whatsapp pueden ser caóticos si de poder consultar información pasada se trata, sino recuerdas bien el tema y no puedes usar el buscador por frases, por otra parte el tener un apartado web sobre el cual poco o nada pueden modificar por cuenta propia no permite la actualización del contenido relevante.

Por consiguiente, presenté la propuesta de desarrollar una web para la AFA que pudiese solventar gran parte de sus necesidades a lo cual no dudaron en aceptar el proyecto y empezar a pensar en los requerimientos que podrían facilitar y agilizar los procesos que realizan diaria, semanal, mensual o por curso lectivo.

La aplicación web a desarrollar permitirá principalmente dar respuesta a las principales inquietudes de las familias en torno a las actividades de carácter social y educativa, la calidad de la educación impartida, el conocimiento de la gestión y control de los fondos que se manejan, así como promover la participación de las familias en la escuela.

La aplicación permitirá a los usuarios, visualizar la información relevante, así como registrarse en las actividades, ser partícipe de las distintas comisiones que tutela la AFA para una activa participación en la escuela. A su vez existirá el lado administrativo de la aplicación, el cual permitirá la gestión y optimización del trabajo. Las funcionalidades que proporciona la aplicación se centran en la gestión de las actividades, permitiendo crear, modificar, asignar y resolver tareas rápidamente. Por último, pero no menos importante, respecto al diseño, este se caracterizará por ser un diseño atractivo que permita que la usabilidad del usuario sea cómoda y fácil. Además, se tratará de un diseño responsive que permitirá su uso desde cualquier dispositivo sea este móvil, Tablet, ordenador.

Para el desarrollo de la aplicación se hará uso de la metodología de entrevistas para la recolección de los datos necesarios que permitan su desarrollo según se mantengan estas entrevistas se crearan las distintas historias que permitirán el avance del desarrollo, así también se usará la metodología ágil para que la AFA tenga una aplicación activa y funcional en cada momento del transcurso del desarrollo, sobre las metodologías comentadas se explicarán con mayor detenimiento en apartados posteriores.

Resum

En aquest treball de fi de grau, es realitzarà una implementació d'una aplicació web que permetrà a l'Associació de Famílies d'Alumnes potenciar la comunicació família-escola.

Com a pare i usuari he observat la necessitat d'un lloc web on poder consultar la informació rebuda de l'Associació de Famílies a qui d'ara endavant cridarem "AFA". Les eines d'informació actuals que utilitza l'AFA són un grup de WhatsApp i un apartat dins d'una web propietat de l'escola que ens és bidireccional la qual s'actualitza amb poca freqüència, com bé se sap els grups de whatsapp poden ser caòtics si de poder consultar informació passada es tracta, sinó recordes bé el tema i no pots usar el cercador per frases, d'altra banda el tenir un apartat web sobre el qual poc o res poden modificar per compte propi no permet l'actualització del contingut rellevant.

Per consegüent, vaig presentar la proposta de desenvolupar una web per a l'AFA que pogués solucionar gran part de les seves necessitats a la qual cosa no van dubtar a acceptar el projecte i començar a pensar en els requeriments que podrien facilitar i agilitzar els processos que realitzen diària, setmanal, mensual o per curs lectiu.

L'aplicació web a desenvolupar permetrà principalment donar resposta a les principals inquietuds de les famílies entorn de les activitats de caràcter social i educativa, la qualitat de l'educació impartida, el coneixement de la gestió i control dels fons que es manegen, així com promoure la participació de les famílies a l'escola.

L'aplicació permetrà als usuaris, visualitzar la informació rellevant, així com registrar-se en les activitats, ser partícip de les diferents comissions que tutela l'AFA per a una activa participació a l'escola. Al seu torn existirà el costat administratiu de l'aplicació, el qual permetrà la gestió i optimització del treball. Les funcionalitats que proporciona l'aplicació se centren en la gestió de les activitats, permetent crear, modificar, assignar i resoldre tasques ràpidament. Finalment, però no menys important, respecte al disseny, aquest es caracteritzarà per ser un disseny atractiu que permeti que la usabilitat de l'usuari sigui còmoda i fàcil. A més, es tractarà d'un disseny responsive que permetrà el seu ús des de qualsevol dispositiu sigui aquest mòbil, Tauleta, ordinador.

Per al desenvolupament de l'aplicació es farà ús de la metodologia d'entrevistes per a la recollida de les dades necessàries que permetin el seu desenvolupament segons es mantinguin aquestes entrevistes es creessin les diferents històries que permetran l'avanç del desenvolupament, així també s'usés la metodologia àgil perquè l'AFA tingui una aplicació activa i funcional a cada moment del transcurs del desenvolupament, sobre les metodologies comentades s'expliquessin amb major deteniment en apartats posteriors.

Abstract

In this final degree project, a web application will be implemented to allow the Association of Student's Families to enhance family-school communication.

As a parent and user, I have seen the need for a website, where I can consult the information received from the Family Association, whom we will call "AFA" from now on. The current information tools used by the AFA are a WhatsApp group and a section within a website owned by the school that is not bidirectional and is updated infrequently. As is well known, WhatsApp groups can be chaotic when there is the need to consult past information, if you do not remember the topic well and you cannot use the search engine by phrases. On the other hand, having a web section on which little or nothing can be modified on your own does not allow to update relevant content.

Therefore, I presented the proposal to develop a website for the AFA that could solve a large part of their needs, to which they did not hesitate to accept the project, and immediately they started thinking about the requirements that could facilitate and speed up the processes that they carry out daily, weekly, monthly or per school year.

The web application to be developed will mainly answer the main concerns of families regarding all the social and educational activities, the quality of the education provided, knowledge of the management and control of the funds, as well as promote the participation of families in the school.

The application will allow users to view relevant information, as well as register themselves for activities, and participate in the different commissions that the AFA manages to promote parental active participation in the school. Moreover, there will be the administrative side of the application, which will allow better management control and optimization of the work. The functionalities provided by the application will focus on the management of activities, allowing you to create, modify, assign and resolve tasks quickly. Last but not least, regarding the design, this will be characterized by being an attractive design that allows the user usability to be comfortable and easy. In addition, it will be a responsive design that will allow its use from any device like a mobile, tablet, or computer.

For the development of the application, the methodology that will be used is the interviews that will allow the collection of the necessary data. Based on how the interviews are maintained, these different stories will enable and create the advancement of this web development. The agile methodology will be used to provide the AFA with a more active and functional application at each moment of the course of development, the methodologies discussed will be explained in greater detail in later sections.

Agradecimiento

Me gustaría agradecer a mi familia por el apoyo y la motivación que he recibido para terminar mi carrera, sin ellos mi esfuerzo no habría merecido la pena.

Agradezco también a mi tutor del TFG, a mis amigos y familias del colegio de mis hijos que me han ayudado a mantener el enfoque, a drenar los problemas para visualizar el proceso y poder llevar este proyecto web a buen término.

Gracias, este trabajo va por todos vosotros.

Contenido

1. Introducción	1
1.1 Objetivos	1
1.2 Antecedentes	2
1.3 Motivación	3
1.4 Estructura de la memoria	4
2. Análisis	5
2.1 Análisis de requerimientos	5
2.2 Identificación del problema	8
2.3 Requerimientos básicos de la aplicación	8
2.4 Metodología	9
2.5 Planificación	11
3. Diseño Web	17
3.1 Patrones de diseño	18
3.2 Modelos diseñados y análisis heurístico	21
3.2.1 Página Principal	21
3.1.2 Páginas noticias y eventos	22
3.1.3 Páginas descripción de servicios	23
4. Desarrollo	25
4.1 Tecnologías utilizadas	25
4.1.1 BootStrap	25
4.1.2 jQuery	26
4.1.3 React	27
4.1.4 Axios	28
4.1.5 Firebase	28
4.1.6 Fontawesome	29
4.1.7 Tinymce	29
4.2 Arquitectura de la aplicación	29
4.2.1 MVC	29
4.2.2 Api Rest	30
4.2.3 Base de datos	30
4.2.4 Esquema web	31
4.2.5 Aplicando patrón de arquitectura MVC	32
4.2.5.1 Modelos	32
4.2.5.2 Vistas	37

4.2.5.3	Controlador	42
4.2.5.3.1	Firebase	42
4.2.5.3.1.1	FileStorage	43
4.2.5.3.1.2	Authentication	43
4.2.5.3.2	Controladores dentro de la aplicación.....	45
5.	Conclusiones	49
6.	Bibliografía	56

1. Introducción

Como bien se sabe, internet es ahora el mundo donde se encuentran muchas de las respuestas a varias inquietudes y búsqueda de conocimiento.

Las Tecnologías de la Información y de la Comunicación juegan un papel fundamental en el escenario educativo: alumnado, profesorado, familia y centro. Dicho esto, las familias no deberían encontrarse con obstáculos para colaborar y comunicarse, al contrario, lo ideal sería la existencia de herramientas que permitan y fomenten la buena comunicación entre familia y escuela a fin de potenciar la participación de todos los agentes para el buen funcionamiento de la asociación y atender las necesidades de los alumnos.

A partir de esto, vemos la necesidad de desarrollar una herramienta de comunicación web para la AFA que permita brindar información relevante creando un espacio idóneo para mejorar la comunicación, facilitar la gestión y optimizar la organización.

1.1 Objetivos

Como objetivo principal está el dar solución a la problemática de visibilidad en internet de la "AFA", poder brindar la accesibilidad a la información relevante para los padres y madres respecto a la asociación y la escuela.

Como segundo objetivo el poder implementar la aplicación que se lleva a cabo en este TFG haciendo uso de los conocimientos adquiridos a lo largo de la curricular de la universidad. Como por ejemplo Software distribuido donde hacemos uso de la diferenciación de las partes de un software para su mejor uso y escalabilidad, así también tenemos Factores Humanos y Computación que nos permite asegurar la accesibilidad, usabilidad y seguridad de los sistemas, también podría mencionar Diseño de Software el cual nos brinda la capacidad de diseñar, desarrollar, selección y evaluar aplicaciones y sistemas informáticos. Estas y otras materias cursadas a lo largo de la carrera profesional de informática.

Como tercer objetivo, el uso de nuevas tecnologías, tecnologías que no he usado, pero con las competencias adquiridas durante el paso por la universidad pueden ser entendibles y que representan un reto, un reto sobre el cual nos brinda todas las herramientas necesarias la universidad en el desarrollo del pensamiento y de la constante búsqueda de conocimiento.

1.2 Antecedentes

La participación de las familias en la UE y de la AELC/EEE

En la participación de las familias en los sistemas educativos de los distintos Estados miembros de la UE y de la AELC/EEE, es interesante subrayar que, en la mayoría de los países, la participación de las familias o de sus representantes en las estructuras organizadas dentro del sistema educativo es una práctica desarrollada principalmente desde 1970. No obstante, en algunos países este proceso comenzó con anterioridad a esta fecha (Alemania, Francia, Luxemburgo, Austria, Finlandia y Noruega). Concretamente la década de 1980 constituye un período muy activo en el establecimiento de un corpus legislativo en la mayoría de los países, salvo en Bélgica (Comunidad francófona), Dinamarca, Italia, Suecia, Islandia y Liechtenstein. Sin embargo, el proceso de participación de las familias no se ha desarrollado de manera lineal en cada uno de los países. Así, hay casos en que han transcurrido alrededor de veinte años entre el primer marco legislativo y el siguiente (Francia, Luxemburgo, Finlandia y Noruega). La autonomía de las escuelas y la participación de las familias en su gestión son dos temas clave en los debates y legislaciones actuales. España ha adoptado, en 1995, una ley (LOPEG) que tiende a reforzar la participación de las familias en el funcionamiento de los centros escolares.

Asociaciones de familias en la UE y de la AELC/EEE

Existen asociaciones de familias y federaciones nacionales en todos los Estados miembros de la UE y de la AELC/EEE. Su estructura no se basa en un único modelo. Su representatividad es diversa, así como su forma de coordinación. En ciertos países, y según los sectores y niveles educativos, las asociaciones de familias son menos frecuentes y activas. En Grecia, España, Austria y Portugal, tales asociaciones están previstas en la legislación escolar. A escala europea, existen federaciones que realizan el trabajo de coordinación: - European Parents Association (EPA). - Confédération des organisations familiales de la Communauté européenne (COFACE). - Groupement international des associations de parents de l'enseignement catholique (OE-GIAPEC)

Legislación AFA

La Ley de Ordenación de la Enseñanza Media, de 27 de febrero de 1953, facultaba al Estado para reconocer la existencia de las que se constituyeran (artículo 70). En los años 1960 se generalizó su implantación en la mayoría de los institutos. El modelo se imitó, vigente ya la Ley General de Educación, de 1970, en la Educación General Básica.

El Real Decreto 1533/1986, con fecha 11 de julio, por el que se pretende regular las AFAs, define, entre otros, las funciones que se deben establecer dentro de las mismas. Las AFAs con intereses similares se pueden unir en federaciones o confederaciones que permiten la presencia de la opinión de las familias y madres ante las autoridades correspondientes.

Dentro de sus órganos de gobierno están las Asambleas Generales, integradas por todos sus asociados, siendo el órgano supremo de la voluntad de la organización. Sus acuerdos son de obligado cumplimiento, y carácter es ordinario y extraordinario

Funciones AFA:

Las funciones de la AFA se dividen en informativas, formativas y de representación

El Real Decreto de 1986 considera primarias las siguientes funciones:

- Asistir y ayudar a todas aquellas familias, madres o tutores que lo requieran, en cualquier asunto que concierna a la educación y a los derechos y deberes de sus hijos, hijas o pupilos.
- Colaborar en las diferentes actividades educativas de los centros escolares. Realizar actividades que consoliden el Proyecto Educativo del Centro.
- Potenciar y desarrollar acciones formativas que promuevan la participación de las familias en el centro escolar y en los órganos de gobierno.
- Promocionar y promover la participación de las familias y madres de los estudiantes en lo que respecta a la gestión del centro.
- Tratar y apoyar adecuadamente a aquellas familias que tengan necesidades pedagógicas específicas o de carácter social.
- Asistir y ayudar a las familias y madres de alumnos que quieran ejercer sus derechos de intervención respecto a la gestión y el control de los centros que se sostienen con fondos públicos.

Disposiciones adicionales según el mismo RD:

- Promover y promocionar la calidad en la educación impartida
- Poner todos los mecanismos necesarios para facilitar la participación y la representación de las familias y madres de alumnos que quieran participar en los Consejos Escolares de centros concertados o públicos, y cualquier otro órgano colegiado
- Trabajar junto al profesorado del centro para asegurar el buen funcionamiento de este
- Implantar mecanismos que fomenten una convivencia de calidad dentro del centro
- Seguir cualquier otra indicación normativa que venga reflejada en los estatutos de la asociación

1.3 Motivación

La principal motivación del desarrollo de este proyecto es la contribución a la agilización y mejora de los canales de comunicación de la AFA.

Cabe resaltar que, aunque la principal motivación sea la mejora de la comunicación entre AFA y familias, también existen otras como la visibilidad web que ganará la AFA al tener una presencia no compartida, sino que su presencia será a partir de la fecha de puesta en marcha una visibilidad propia. Así también se verá un crecimiento en la interacción entre las familias y el centro educativo como un efecto secundario al tener las familias un lugar alternativo donde consultar la información relevante respecto a los eventos y noticias del centro y de la AFA. La apertura de este nuevo canal traerá consigo muchos beneficios.

Adicionalmente a lo comentado también existe la motivación del conocimiento. Como se indica en este TFG se tocarán nuevas tecnologías que tienen un gran avance y que

proviene de grandes empresas. Como Firebase de Google que con su SDK permite un desarrollo más efectivo o React que fue desarrollada por Facebook y que permite ese desarrollo de interfaz de usuario reactivo.

1.4 Estructura de la memoria

La memoria está estructurada en forma de capítulos con las diferentes fases y partes del proyecto. A continuación, se procede a su definición.

Introducción

En este apartado se define una breve introducción al desarrollo de este TFG y los apartados objetivos, antecedentes y la motivación de este TFG.

Análisis

En este apartado se realiza el análisis correspondiente que permitirá decidir los requerimientos de la aplicación que se desarrolla en este TFG. Apartados que intervienen en este capítulo son análisis de requerimientos, identificación del problema, requerimientos básicos de la aplicación, metodología y planificación.

Diseño web

En este apartado primero se realiza una breve definición del diseño y se mencionan algunos patrones de diseño y reglas heurísticas que permitan un diseño eficaz de la interfaz web a desarrollar para la aplicación de este TFG. Entre los apartados incluidos se tiene patrones de diseño y modelos diseñados, así como el análisis heurístico de los diseños.

Desarrollo

En este apartado se define una breve introducción al desarrollo de este TFG y los apartados objetivos, antecedentes y la motivación de este TFG.

Conclusiones

En este apartado se define una breve introducción al desarrollo de este TFG y los apartados objetivos, antecedentes y la motivación de este TFG.

Bibliografía

En este apartado se define una breve introducción al desarrollo de este TFG y los apartados objetivos, antecedentes y la motivación de este TFG.

2. Análisis

En este apartado de análisis se observarán los requisitos previos solicitados para la AFA que sería el cliente que desea obtener la aplicación. También se analizarán los distintos factores externos que tienen importancia para la aplicación, como son los grupos de usuarios que utilizarán la aplicación (padres, madres, niños, público en general). Basándonos en estos detalles extraeremos conclusiones que serán útiles para la planificación y el diseño de las funcionalidades.

2.1 Análisis de requerimientos

En este apartado se hará en primera instancia una toma de requerimientos inicial, esta toma de requerimientos se hará haciendo uso de entrevistas, en las cuales se ha intentado escuchar y mantener todo lo que los demandantes de la aplicación han querido explicar sobre el producto que desean obtener al final del desarrollo.

A continuación, se presenta una de las entrevistas desarrolladas.

¿Qué es lo que necesita?

Una página web para comunicarme con los padres de familia

¿Porque desea comunicarse con los padres de familia?

Porque actualmente el único medio de comunicación es un grupo de whatsapp y un apartado dentro de la web de la escuela, gestionado por la escuela. Requiriéramos un sitio donde se pueda visualizar y compartir la información con los padres.

¿Qué tipo de información es la que desean compartir con los padres de familia?

Noticias relevantes de la escuela

Eventos escolares

Eventos de la comunidad

Compartir todos los servicios que ofrecemos como son:

Extraescolares, Comedor, Equipamiento, Jornadas para padres.

Compartir información sobre las distintas asambleas de padres realizadas durante el año escolar.

Compartir información de contactos para la atención a la familia.

¿Qué problemas tienen con los medios de comunicación actual?

Whatsapp:

La información dentro de los grupos de whatsapp y si el usuario lo elimina no mantiene la información que hemos pasado con anterioridad.

No todas las personas que deberían de tener la información están en el grupo de whatsapp porque son tantos que no siempre están unidos al grupo.

En los grupos de whatsapp no solo se agrega contenido de la AFA o Escuela, pueden existir distintos mensajes que conviertan en un caos el grupo, olvidando el fin para el que fue creado.

Apartado Web.

La estructura de la información es confusa.

No existe libertad para manejar el contenido a tiempo real, por ser manejado por la escuela.

No se puede agregar mucho contenido porque solo es un apartado dentro de la web principal de la escuela.

¿A quién va dirigida esta Web, o quienes la usarán?

Va dirigida a padres, madres y tutores legales, público en general que esté interesado en las actividades que realizamos.

Profesores.

Usuario Administrador de la AFA.

¿Qué es la AFA?

Es la Asociación de Familias, que gestiona y brinda asesoramiento e información a los padres y madres de familia.

¿Cuántas personas cree usted que consultarán esta página?

700 - 1000

¿Qué debería de hacer la página web para que usted diga que le sirve?

Que cargue rápido,

Que presente la información que quiero proporcionar,

Que sea fácil de encontrar la información.

Que se pueda ver en cualquier dispositivo

¿Desea usted mantener el contenido de la página web?

Si, entonces se ha de realizar una administración para la página web.

¿Qué necesita administrar dentro del contenido?

Poder agregar, eliminar o modificar una extraescolar

Poder agregar, eliminar o modificar un evento

Poder agregar, eliminar o modificar una noticia

Poder agregar, eliminar o modificar el contenido del apartado del comedor

Poder agregar, eliminar o modificar el contenido de actas, estatuto y presupuestos

Poder agregar, eliminar o modificar el contenido de la presentación de la AFA

Poder consultar cuantos inscritos hay por extraescolar

Poder consultar cuantos inscritos hay en cada comisión

Poder consultar cuantos inscritos nuevos asociados en la AFA existen

Poder consultar cuantos mensajes de consultas existen o las consultas realizadas a la AFA.

¿Gestionará algún tipo de pago en la web?

No.

¿Necesitará algún formulario?

Si, Varios.

¿Me podría indicar que formularios?

Consultas

Inscripción a extraescolares

Inscripción a comisiones.

Inscripción a la AFA

2.2 Identificación del problema

No existe un canal propio que permita mantener la información presente en todo momento para los usuarios que requieren la información que brinda la AFA.

El apartado web es insuficiente para poder desplegar todo el contenido que puede brindar la AFA a los asociados y público en general

Los grupos de whatsapp no son suficiente para brindar la información relevante y mantenerla disponible, asequible para todos los usuarios que requieren de dicha información.

No se puede llevar un control satisfactorio sobre los asociados, inscritos a las comisiones y extraescolares que mantienen como servicio la AFA.

2.3 Requerimientos básicos de la aplicación

Una vez realizado un primer análisis usando como medio de obtención de los datos entrevistas realizadas a dos grupos fundamentales que usarán la aplicación “Familias, Administradores AFA” se han podido identificar los siguientes requerimientos básicos para que la aplicación sea funcional.

Requerimientos Operacionales

Visualizar información de servicios

Visualizar extraescolares

Poder anotarse a una extraescolar

Poder anotarse en una comisión

Poder afiliarse a la AFA

Poder enviar consultas a la AFA

Visualizar información sobre Eventos

Visualizar información sobre el comedor de la escuela

Visualizar información sobre lo que es la AFA

Visualizar información de transparencia de la AFA

Modulo administrativo

Realizar login administrativo

Agregar, eliminar, modificar Extraescolar

Realizar seguimiento de las extraescolares

Agregar, eliminar, modificar eventos

Agregar, eliminar, modificar noticias

Agregar, eliminar, modificar contenido de los apartados de la web.

Consultar información relevante sobre los anotados a las extraescolares, comisiones, nuevos afiliados, etc.

Requerimientos no funcionales

Rápido acceso a los apartados de la web

Diseño moderno de la web

Que la web se pueda visualizar en distintos dispositivos

Una vez identificados los requerimientos básicos de la aplicación se ha realizado un breve estudio de la mejor manera de afrontar el problema, de que manera dar un valor funcional constante durante el desarrollo. Que metodología permitiría que la AFA pueda tener un producto entregable en cada momento del desarrollo. Pues es muy necesario para ellos que se puedan ir observando los avances y que los usuarios puedan ir viendo en cada instante que la web que se está desarrollando va entregando el contenido necesario con la información relevante para las familias.

Dicho todo lo anterior se he definido el uso de las metodologías ágiles haciendo uso del marco estructurado SCRUM, el cual me permitirá siempre poder otorgar ese valor funcional después de cada iteración.

2.4 Metodología

En el desarrollo de la aplicación se ha decidido el uso de las metodologías ágiles, basándonos en el marco de trabajo SCRUM, el cual nos permite un trabajo colaborativo. Aunque si bien es cierto el grupo de desarrollo este compuesto por una sola persona, aun así, he creído necesario usar esta metodología por el beneficio principal de poder otorgar un producto con valor al final de cada sprint.

SCRUM no solo permite brindar este valor, sino también facilita el aprendizaje continuo en la adaptación a todo factor fluctuante. Lo cual permite la adaptación a los cambios en los requisitos conforme el desarrollo de la aplicación va avanzando. Aunque SCRUM es estructurado, no es del todo rígido, su ejecución se pueda adaptar a las necesidades de cualquier organización.

Los principios básicos de Scrum son la innovación, la competitividad, la flexibilidad y sobre todo la productividad y la unidad de los equipos.

Como he mencionado, la metodología ágil y en lo particular el uso del marco de trabajo SCRUM se desarrolla de forma iterativa e incremental, cada iteración se denomina SPRINT.

Estas iteraciones pueden durar entre 2 o 4 semanas, aunque lo importante es que no lleven más de dos semanas en cada iteración porque si lo estuviésemos haciendo, es porque nos estaríamos enfrentado a lo que se llama o denomina una Epic story, lo cual permitiría

subdividir esta story para poder realizar estas subdivisiones en lo propuesto de no sobrepasar dos semanas de desarrollo por cada sprint, este ajuste permite el obtener un valor agregado al final de cada sprint.

A continuación, breve explicación de las partes de un sprint.

- **Product Backlog:** Este es el conjunto de historias de usuario, las cuales son definidas en lenguaje simple y el contiene el valor de una funcionalidad o requisito que se pide en el desarrollo de la nueva aplicación.
- **Sprint Planning:** Aquí nos enfrentamos a la elección de las historias a desarrollar en el sprint, las cuales han sido valoradas por el product owner por prioridad y valor dentro del contexto del desarrollo de la aplicación. En este momento los desarrolladores darán una primera valoración a cada story para poder determinar el tiempo de desarrollo del sprint.
- **Sprint:** El Sprint como se indica es el momento de duración determinada con anterioridad donde se procede a completar y/o desarrollar las historias asignadas al mismo. Después de este sprint se tendrá una aplicación funcional de acuerdo a las historias desarrolladas.
- **Backlog:** Son las historias que se desarrollarán durante el sprint.
- **Daily meeting:** Reunión que tendrá un máximo de 15 minutos donde cada componente del equipo de desarrollo indicará el estado de las historias que tiene a su cargo en el desarrollo del sprint y el tiempo estimado que le queda para la finalización de cada una de ellas. Si existiese algún impedimento para alguna de las tareas, será en esta reunión donde deberá indicarlo para poder realizar la reorganización necesaria para poder tener el desarrollo del sprint de acuerdo con fecha determinada con anterioridad.
- **Demo y retrospectiva:** En este momento es cuando el sprint ha finalizado y en el cual se realiza la presentación del sprint finalizado, se presentan las historias acabadas y las no acabadas para poder hacer una retrospectiva sobre el desarrollo y así poder afrontar el siguiente sprint con el refuerzo del aprendizaje de todo lo ocurrido en el sprint. Así como también si existiera algunas historias no realizadas se pueda resolver si por recursos o porque la historia no fue totalmente explicativa o porque fuese una epic story se deba subdividir o asignar en un futuro sprint con mayores recursos.

2.5 Planificación

Para obtener una aproximación de la distribución del tiempo en las tareas del proyecto, se crearon historias de usuario a partir de los requerimientos básicos que puedan ser divisibles en sprint los cuales permitirán a su vez la realización satisfactoria de la aplicación. Así también se realizó el diagrama de Gantt con el objetivo de gestionar y planificar el tiempo de dedicación previsto en cada actividad.

División de Sprint:

SPRINT 1:

Historia 1: quisiéramos ver un menú con los apartados necesarios que puedan contener la información que queremos presentar.

Historia 2: quisiéramos poder ver imágenes del AFA de manera que se presenten una a una.

Historia 3: quisiéramos que hubiese otra forma de ingresar a los apartados no solo el menú principal.

SPRINT 2:

Historia 4: quisiéramos que en el menú AFA hubiese un apartado de Quienes somos.

Historia 5: quisiéramos que en el menú AFA hubiese un apartado de comisiones

Historia 6: quisiéramos que en el menú AFA hubiese un apartado para hacerse socio de la AFA.

SPRINT 3:

Historia 7: quisiéramos un formulario de contacto.

Historia 8: quisiéramos que en el apartado transparencia exista un menú donde se pueda ver los estatutos de la AFA.

Historia 9: quisiéramos que en el apartado transparencia exista un menú que permita visualizar todos los presupuestos anuales.

SPRINT 4:

Historia 10: quisiéramos que en el apartado transparencia exista un menú que permita visualizar las actas del año.

Historia 11: quisiéramos que, en el apartado servicios, donde planteamos los servicios que ofrecemos exista un menú que permita ver la información del comedor de la escuela.

Historia 12: quisiéramos que en el apartado servicios también se incluya un menú en el cual se pueda ver la información de la acogida que ofrecemos para la conciliación de la vida laboral de las familias y la escuela de sus hijos.

SPRINT 5:

Historia 13: quisiéramos que dentro del apartado de servicios exista una página donde podamos presentar la información que desarrollamos para las tardes de junio donde la escuela finaliza antes de las 4:30 o 5 para poder conciliar la vida laboral de los tutores legales con la vida escolar de sus hijos.

Historia 14: quisiéramos una página de equipamiento, donde podamos agregar la información relevante del equipamiento anualmente.

Historia 15: quisiéramos una página que contenga la información del casal de verano que se desarrolla tanto en las instalaciones de la escuela como en instalaciones de lugares con los cuales se ha realizado algún convenio para el desarrollo de estas.

SPRINT 6:

Historia 16: quisiéramos poder tener una página donde se presenten las extraescolares que planeamos desarrollar durante el año curricular para los alumnos, esto nos parece que debería de ir en el apartado de servicios.

Historia 17: quisiéramos una página que contenga la información de los eventos anuales.

Historia 18: quisiéramos una página que contenga la información de las noticias que comparta la escuela o la AFA a título propio.

SPRINT 7:

Historia 19: quisiéramos poder tener una parte administrativa para en un futuro poder hacer algunas tareas internas en ella.

Historia 20: quisiéramos poder realizar el acceso a dicha área mediante un login usando un mail que todos tenemos, por lo pronto será solo un único usuario.

Historia 21: quisiéramos poder tener dentro del área administrativa un apartado que permita la generación y/o modificación del contenido del apartado comedor.

SPRINT 8:

Historia 22: quisiéramos poder tener dentro del área administrativa un apartado que nos permita la generación de eventos que se desarrollan en la escuela.

Historia 23: nos gustaría poder tener un apartado dentro del área administrativa que permita modificar o crear noticias que se desplegaran en la página de noticias.

Historia 24: nos gustaría poder tener dentro del área administrativa un lugar donde poder crear los eventos, editarlos o eliminarlos para que se presentan dentro del apartado de eventos en la página general de la web.

SPRINT 9:

Historia 25: nos sería útil dentro del área administrativa poder agregar comisiones, así como también poder eliminarlas o editarlas.

Historia 26: en el área administrativa nos sería de utilidad poder nosotros agregar, modificar o eliminar las extraescolares que presentaremos cada año curricular.

Historia 27: nos sería de utilidad que desde el área administrativa podamos modificar o agregar contenido para el apartado acogida de la web.

SPRINT 10:

Historia 28: como parte del área administrativa nos sería útil tener un apartado que nos permita modificar o agregar contenido al apartado de la web tardes de junio.

Historia 29: dentro del área administrativa sería útil tener un lugar desde donde modificar o agregar contenido nuevo a la página de casal de verano.

Historia 30: sería de utilidad para nosotras tener un apartado dentro del área administrativa que permita agregar contenido o editar el contenido de la página equipamiento.

SPRINT 11:

Historia 31: poder modificar el contenido o agregar nuevo contenido a la página actas desde el área administrativa.

Historia 32: tener apartado dentro del área administrativa para poder modificar o agregar nuevo contenido a la página de presupuestos.

Historia 33: tener un formulario de inscripción a las comisiones para nuevos afiliados a las comisiones

SPRINT 12:

Historia 34: nos sería de utilidad para nosotros poder tener un apartado para poder editar y crear contenido para la página de estatutos.

Historia 35: cuando entre al apartado administrativo quisiera poder agregar el contenido de la página Que somos.

Historia 36: tener un formulario de inscripción a las extraescolares para los alumnos.

SPRINT 13:

Historia 37: quisiera poder desde el área administrativa poder modificar el contenido de la página donde las familias pueden hacerse socios de la AFA. Modificar el prólogo que describe porque es bueno pertenecer a la AFA.

Historia 38: tener un formulario de inscripción a la AFA para nuevos afiliados.

Historia 39: nos agradecería que la página se pueda ver en todo tipo de dispositivos porque no siempre se tiene un ordenador a la mano, pero si un móvil o una tablet.

SPRINT 14:

Historia 40: quisiera poder tener dentro del área administrativa un lugar donde poder ver los mails que nos han enviado y así poder tener todo en un solo lugar donde pueda consultar esta información.

Historia 41: nos gustaría poder tener un apartado dentro del área administrativa desde donde pueda consultar los inscritos en las distintas comisiones.

Historia 42: sería de utilidad poder tener un lugar dentro del área administrativa desde donde poder consultar los afiliados a la AFA.

SPRINT 15:

Historia 43: nos ayudaría tener en el área administrativa un apartado que permita ver la información de los alumnos que se anotan en las extraescolares.

Historia 44: cuando se inscriban en una extraescolar se ha de tener siempre las plazas que queden en todo momento libres.

Historia 45: dentro del área administrativa nos sería útil el poder tener un lugar desde donde poder hacer un mailing a todas las familias afiliadas al AFA.

SPRINT 16:

Historia 46: Nos agradecería poder adjuntar imágenes de los eventos realizados en las noticias que presentamos en el apartado noticias al darle un clic a la noticia.

Historia 47: poder modificar o agregar más imágenes al carrusel de la web de la AFA.

Historia 48: poder tener un enlace al Instagram de la escuela.

SPRINT 17:

Historia 49: poder agregar más usuarios con determinados perfiles que puedan tener funciones dentro de la web.

Historia 50: poder realizar consultas de los datos de las extraescolares como por ejemplo ¿Qué extraescolar es la más demandada?

Historia 51: almacenar datos de los usuarios que no han podido obtener una plaza en alguna extraescolar para poder tener encuentra la creación de nueva extraescolar bajo la misma temática.

SPRINT 18:

Historia 52: poder exportar los resultados de las consultas a formato xls.

Historia 53: tener un formulario para que los tutores puedan enviarnos como datos adjuntos los pagos al banco para la inscripción de la extraescolar y así poder tener almacenado este comprobante dentro de la inscripción.

Historia 54: poder tener la posibilidad de agregar y/o modificar el menú principal.

Diagrama de Gantt Provisional

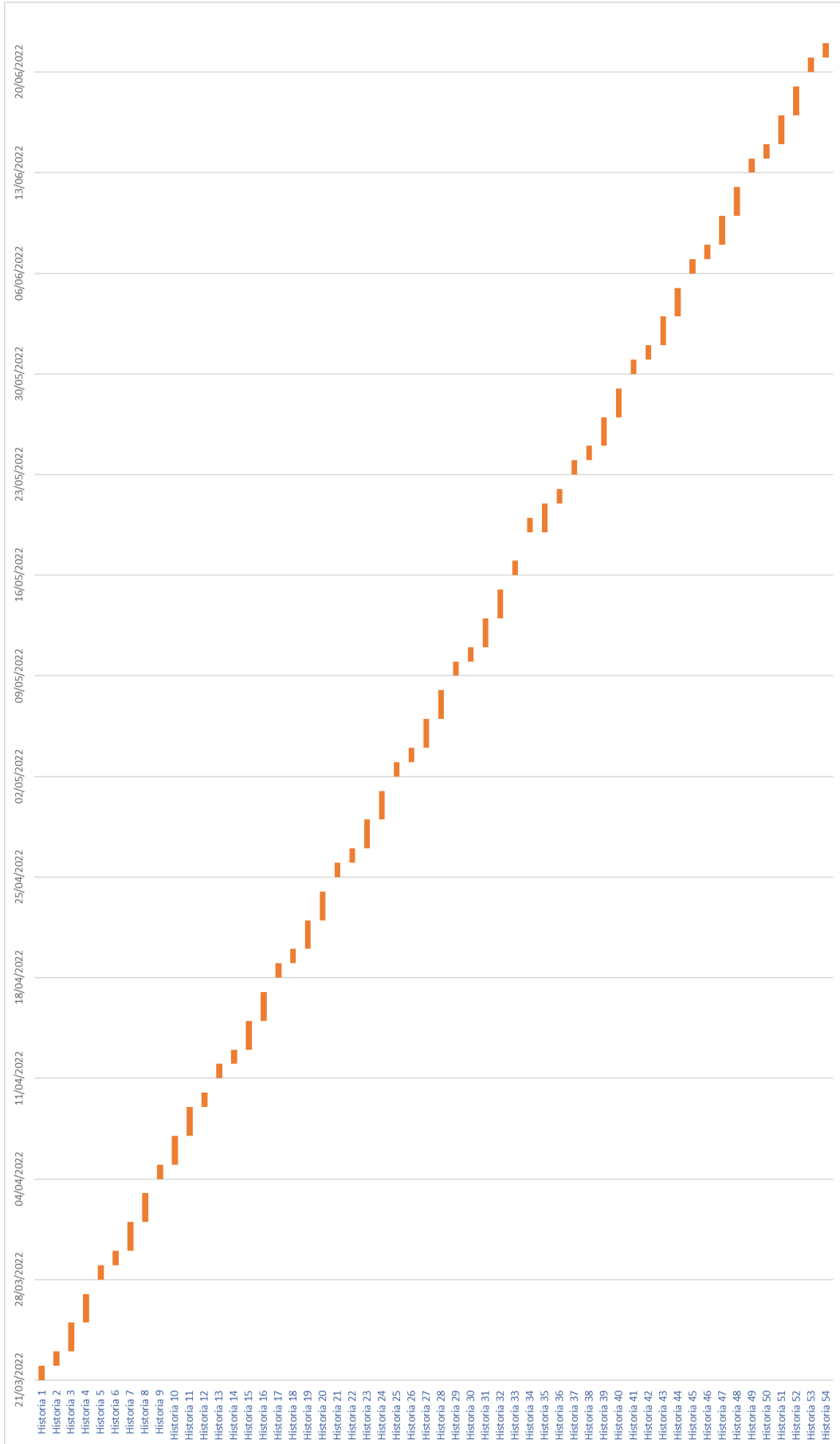
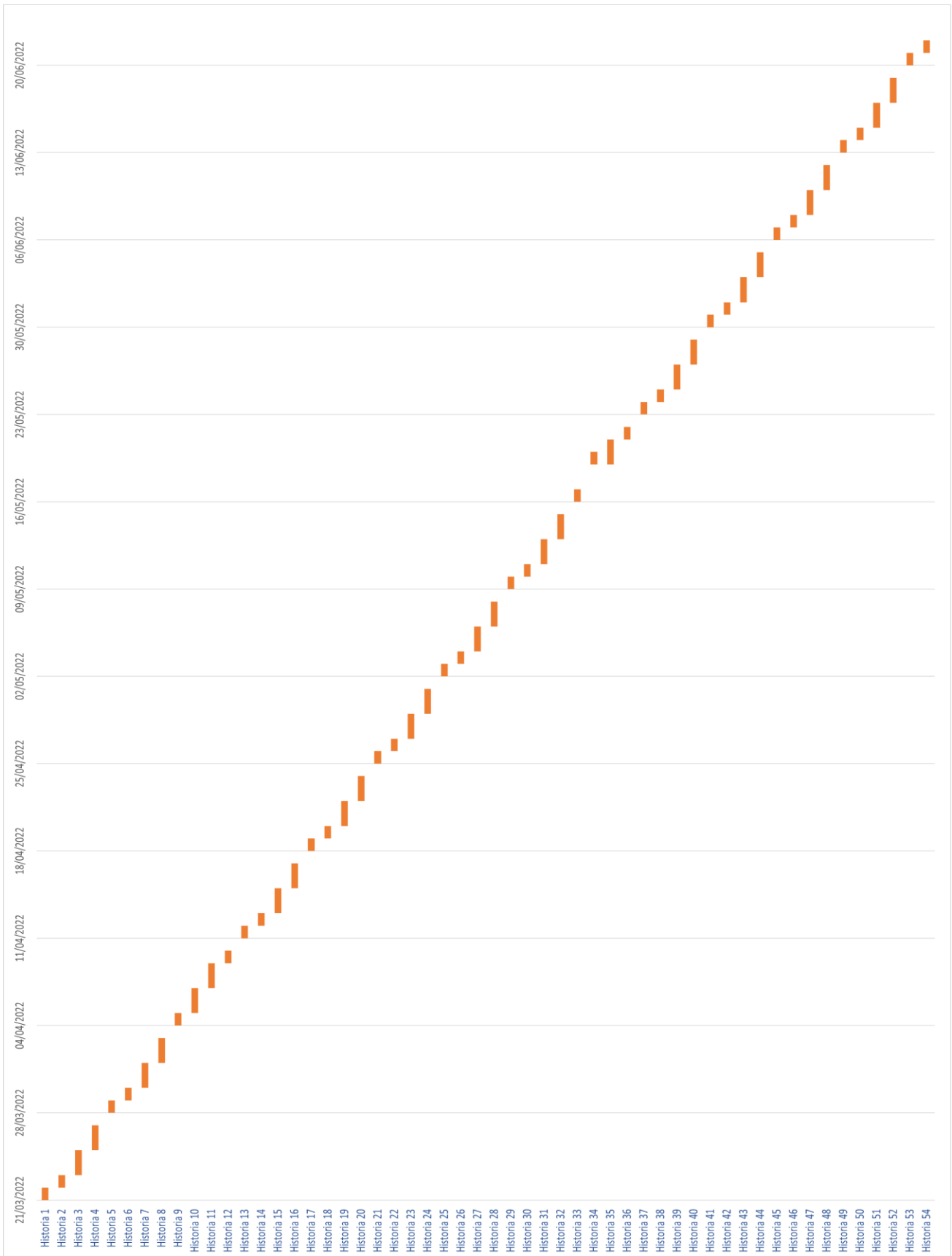


Diagrama de Gantt después del desarrollo



3. Diseño Web

Para el diseño eficaz de una interfaz y/o aplicación web es necesario el diseño de un esbozo conceptual que contenga las funcionalidades y posicionamiento de los elementos para ser mostrados como idea general de la implementación de la interfaz web.

Para el caso que se trata en este TFG se hará uso de algunas de las heurísticas de Jakob Nielsen, por lo que se mencionaran y definirían a continuación.

1. Visibilidad del estado del sistema ¿Qué está pasando?

El usuario debe saber en todo momento que está pasando, el sistema ha de informar de lo que está haciendo, proveer de un feedback al usuario para que sepa que parte de la web está visitando y que pasos ha seguido para llegar al estado en el que se encuentra. Si estuviese el usuario realizando una compra, el usuario deberá poder saber cuántos pasos le quedan para concluir o cuantos pasos ha realizado para llegar al estado en el que se encuentra.

2. Conexión entre el sistema y el mundo real

El sistema y el usuario tienen que hablar en el mismo idioma, siguiendo las convenciones del entorno del usuario.

3. Uso y control del usuario

La imperfección del ser humano permite siempre errar en casos en los que las personas desconocen algo. Dicho lo anterior cuando una persona se enfrenta a un sistema informático siempre lo verá con recelo a la hora de interactuar con él por el miedo a cometer errores en el mismo. Es por ello que se ha de permitir que el usuario tenga un control suficiente de la aplicación que le permita deshacer el error cometido. Que el sistema pueda prevenir acciones que permitan al usuario poder sentir que el control es propio de la persona y no del sistema.

4. Consistencia y estándares

Es importante que en el desarrollo de la aplicación o interfaz exista consistencia, que una acción similar mantenga similitud en todo momento y lugar dentro de la interfaz para que el usuario tenga el conocimiento adquirido de haber realizado acción similar como por ejemplo si existe un botón de aceptar en dos apartados, dicho botón haga exactamente la misma acción, aceptar una validación o aceptar un consentimiento, pero no así uno un consentimiento y el otro eliminar un objeto de la compra. Por otro lado, es necesario el uso de estándares que permitan fluides en la interfaz diseñada.

5. Prevención de errores

Como hemos definido el usuario o las personas cometemos errores, pero si estos errores pueden ser previstos con anterioridad definiendo con claridad la utilidad de cada pantalla, cada elemento dentro de la interfaz, esto permitirá prever de manera eficaz estos errores.

6. Reconocer mejor que recordar

El usuario dispone de poca memoria a corto plazo, por lo que minimiza el uso de su memoria colocando las opciones a la vista o de fácil acceso. Es mucho más fácil que reconozca algo a que lo recuerde estando muy ligado al punto 4 de Consistencia y estándares.

7. Uso eficiente y flexibilidad

El diseño debe servir tanto para usuarios inexpertos como expertos.

8. Diseño práctico y minimalista

No todas las acciones pueden estar a la vista, ya que cada unidad de información reduce la visibilidad de la información que de verdad importa. Manejar correctamente las jerarquías visuales y el espacio en blanco.

9. Ayuda, diagnóstico y recuperación de errores

En el caso de que haya una situación de error, el sistema debe indicar al usuario que ha pasado y cómo resolverlo.

10. Ayuda y documentación

Aunque los sistemas desarrollados puedan ser lo suficientemente usables y no sea necesaria ninguna documentación, siempre habrá usuarios que la requieran. Por esto será importante verificar que el sistema ofrezca ayuda relevante en el contexto que se encuentre el usuario.

Así como hemos querido hacer uso de las 10 heurísticas de Jakob se ha querido basar la elección del patrón de diseño web a usar en los siguientes dos puntos claves.

- Ajustar tu contenido: el diseño web de una página debe apoyar el mensaje que deseas transmitir. Algunos patrones son mejores para mostrar productos, mientras que otros son preferibles para transmitir información rápida y eficazmente (un blog o un portal de noticias). El patrón de diseño web que elijas debe ajustarse a tu tipo de contenido.
- Usar patrones estándar: aunque admiten cierta creatividad, estos patrones de diseño web son generalmente la mejor opción porque han demostrado una y mil veces que funcionan. Como son muy comunes, cumplen las expectativas del usuario que ya está acostumbrado a ellos son más intuitivos y fáciles de usar.

3.1 Patrones de diseño

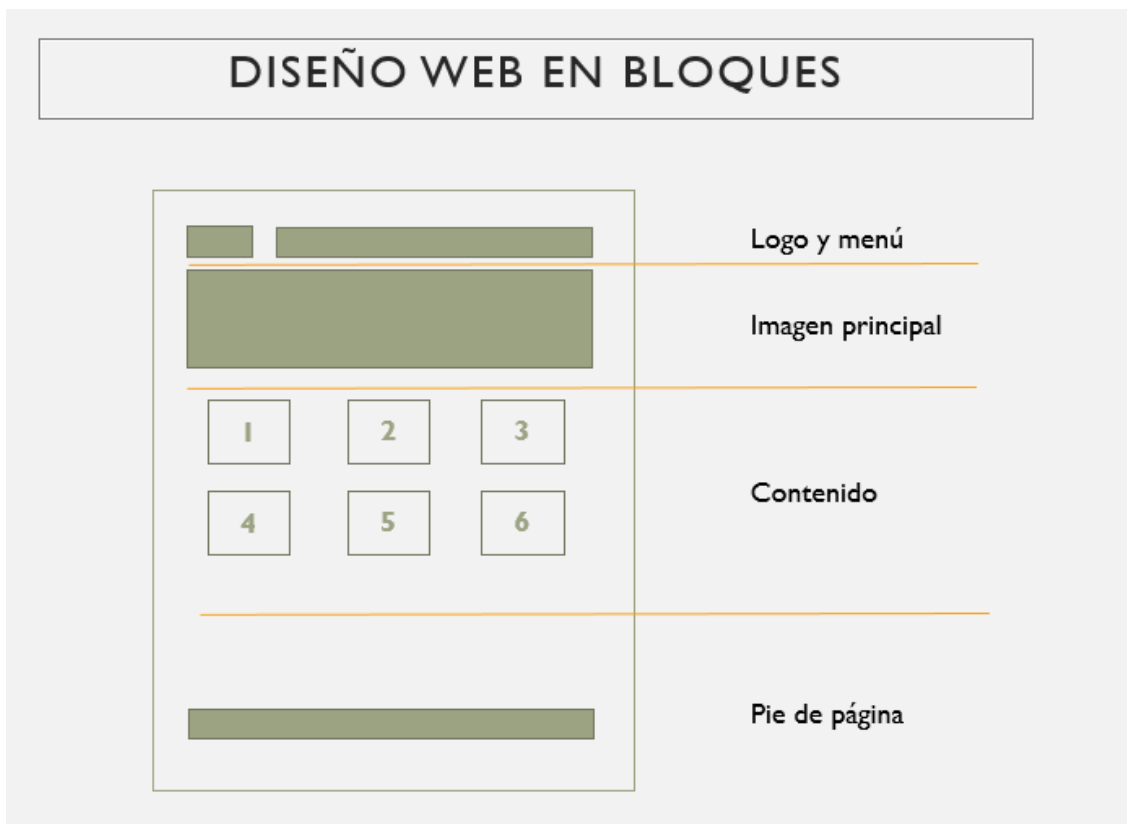
En esta sección se hará un breve resumen de los patrones sobre los cuales se fundamentará el desarrollo de la interfaz web.

Diseño web con patrón de bloques

El diseño web en bloques combina muchas piezas de contenido en un solo diseño geométrico. Gracias a la hábil distribución de la información en cada bloque, los elementos no se solapan entre ellos, sino que contribuyen a dar uniformidad al look general del diseño.

En este patrón, cada bloque conduce a una página interior distinta donde el usuario puede acceder a más información sobre el tema que más le interese.

Ideal para páginas web ricas en contenido. Este patrón de diseño web es una buena solución para mostrar información en bloques que conducen a un apartado distinto a los demás.

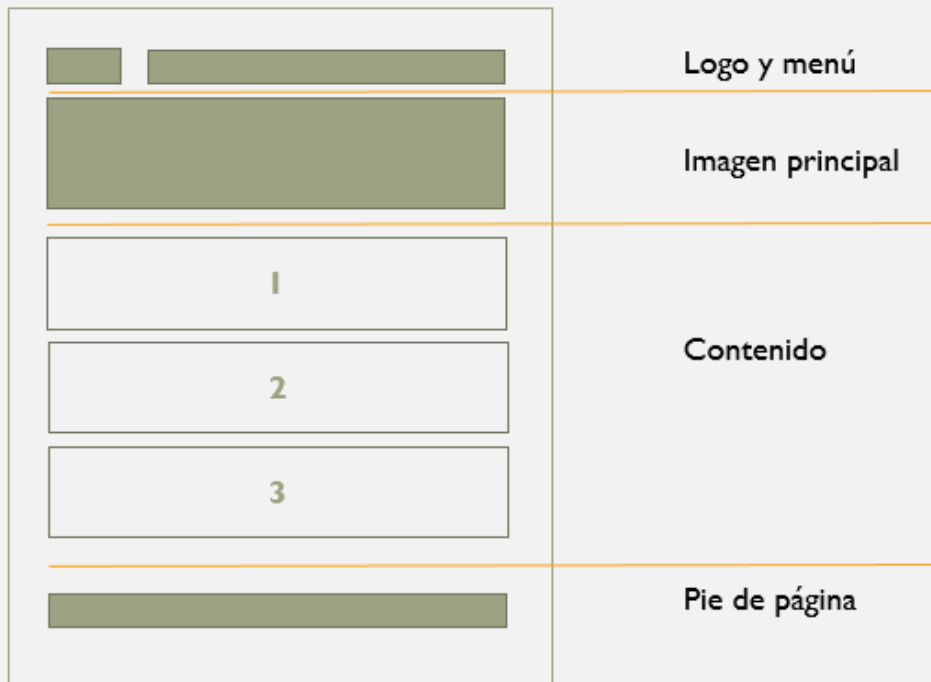


Diseño web de columna única

Este patrón de diseño web presenta todo su contenido en una columna vertical. Es un diseño sencillo y directo (de hecho, es el que usa esta página).

La navegación aquí es fácil de entender, basta con hacer *scroll* hacia abajo para obtener más información. Sin embargo, en este tipo de diseño hay un consejo sobre navegación web que conviene seguir: agregar siempre un botón 'Volver al principio' o un menú flotante fijo para ayudar al usuario a explorar mejor la página.

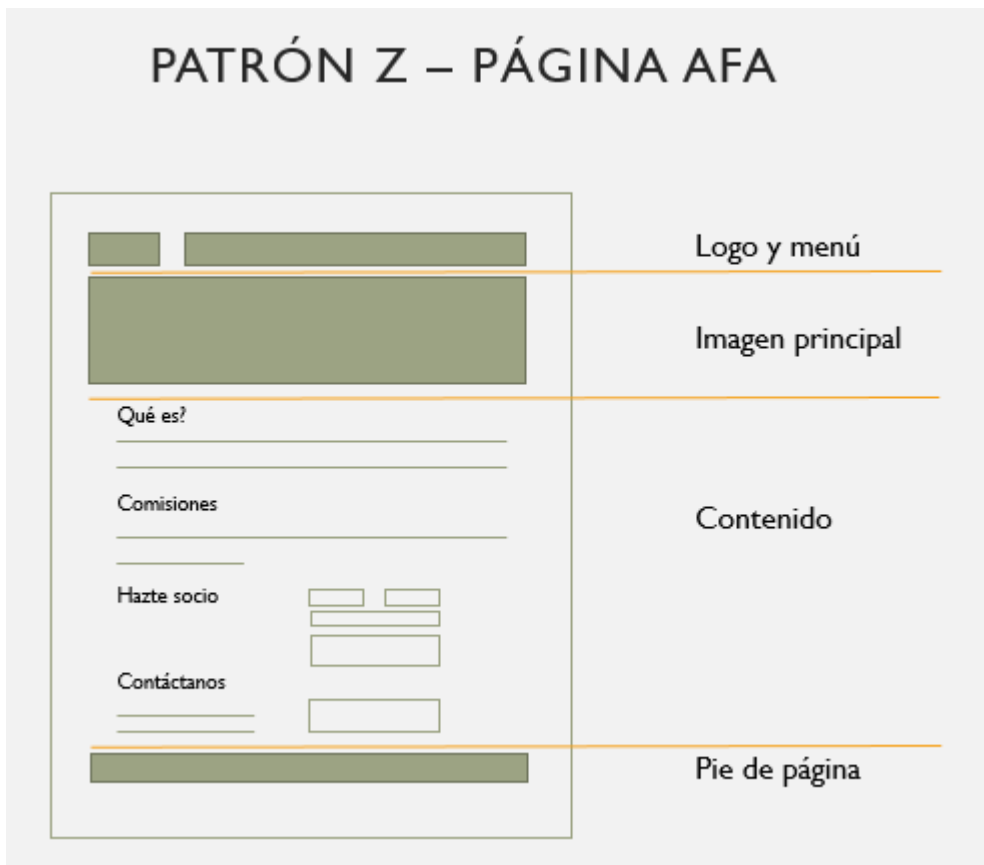
PATRÓN COLUMNA UNICA



Se ha basado el desarrollo de la interfaz web en los dos patrones de diseño porque se realizará una mezcla de estos en toda la interfaz web. Pues en ciertos apartados como extraescolares es conveniente poder presentar la información y la inscripción en la misma página para que así el usuario puede tener las extraescolares y su inscripción de forma relevante en la misma página y sobre el uso del patrón de bloques porque, así como en lo anterior se puede encontrar en la parte administrativa un layout el cual se subdividirá en bloques para poder tener una mejor navegabilidad por parte del usuario administrador.

A continuación, se presenta una muestra de un prototipo que se presentó a la AFA para su aprobación, el cual fue rechazado porque no cumplía con la heurística de Nielsen de un diseño práctico y minimalista al tener todos los apartados en una misma página.

PATRÓN Z – PÁGINA AFA



Se presentaron adicionalmente los siguientes bocetos y/o prototipos.

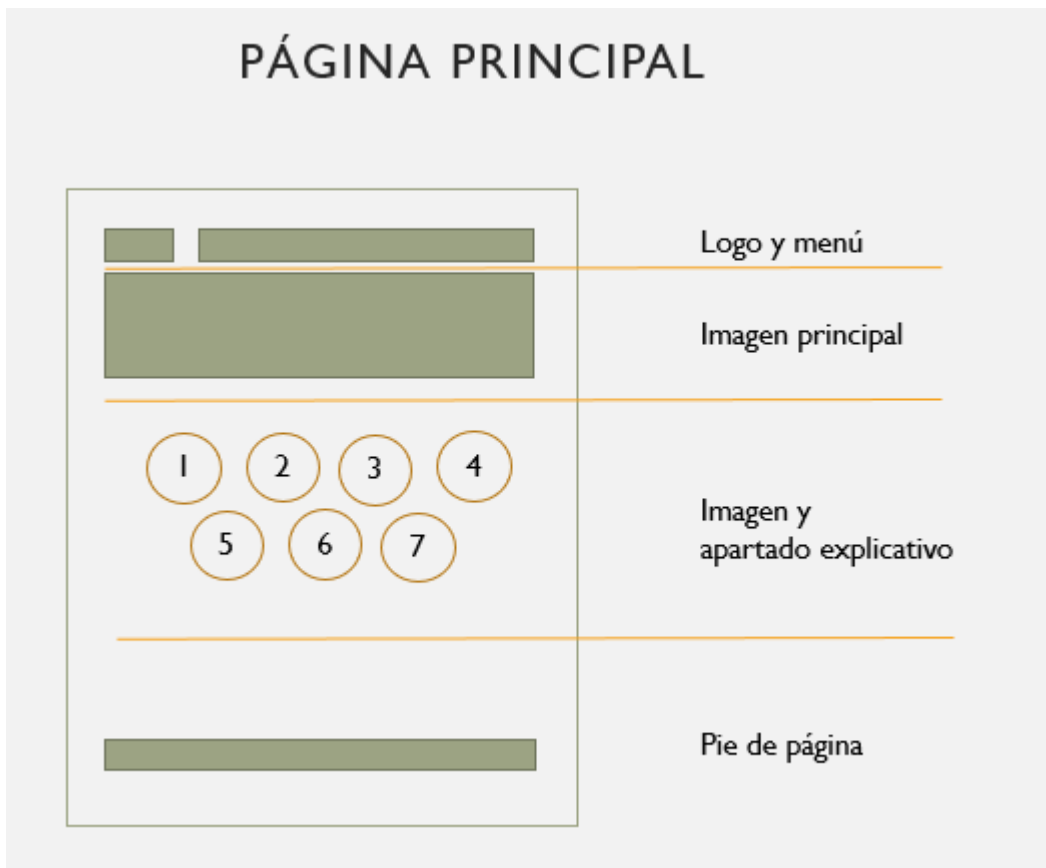
Los cuales fueron aprobados por cumplir con los patrones anteriormente seleccionados. Una mezcla entre el patrón de columna única y el patrón por bloques.

3.2 Modelos diseñados y análisis heurístico

Esta sección presenta algunos diseños realizados, los cuales serán analizados utilizando las heurísticas de Nielsen.

3.2.1 Página Principal

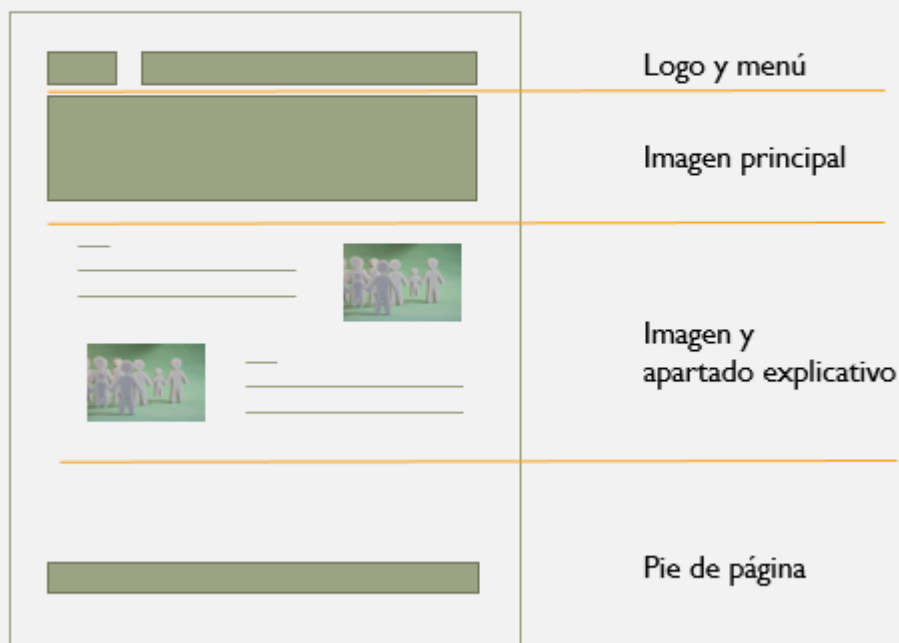
Este primer diseño como se indica corresponde conceptualmente a la página principal. La información se ha organizado en tres secciones. Menú, cabecera, cuerpo y pie de página. En el menú, concretamente se encuentra el logotipo de la aplicación, en él se aplicó la heurística de “Uso y control del usuario” para solucionar la necesidad de un usuario de querer volver a visualizar la página de principal si así lo desea, la manera de retroceder al principio. A si también se hace uso de las heurísticas de “Consistencia y estándares”, para mantener la consistencia en el diseño y en otros diseños, el menú se mantiene en la parte superior de la pantalla para que, de esta forma, el usuario pueda acceder fácilmente. En la sección del cuerpo se ha organizado usando la heurística de “Diseño práctico y minimalista” para mantener una concordancia entre el contenido del cuerpo y el espacio en blanco.



3.1.2 Páginas noticias y eventos

Este segundo diseño corresponde conceptualmente a las páginas Eventos y Noticias que contendrán información a presentar con las noticias relevantes y eventos realizados y a realizar durante un año curricular escolar. Como se indicó con anterioridad el menú principal es heredado y por ello es una de las secciones comunes tanto en la página principal como en las demás páginas creadas para la aplicación web. Con esto usamos el principio heurístico de “Reconocer mejor que recordar”, respecto a la sección del cuerpo se aplica el principio heurístico de “Diseño práctico y minimalista”. Así también en este tipo de páginas se usa el diseño de bloques, al permitir el acceso a galerías distintas de imágenes de acuerdo con la respectiva noticia o evento seleccionado.

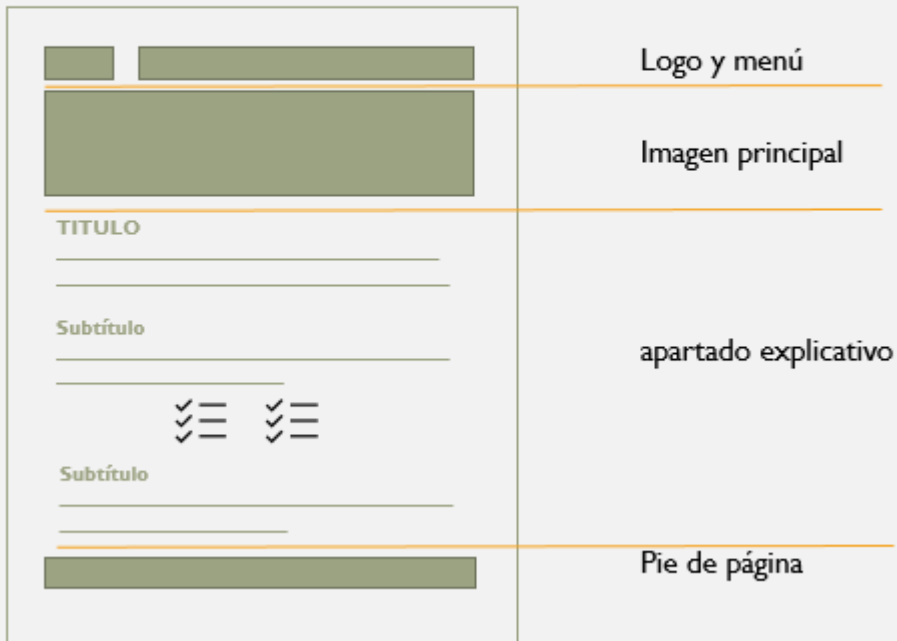
PÁGINAS NOTICIAS Y EVENTOS



3.1.3 Páginas descripción de servicios

Este tercer diseño corresponde conceptualmente a las páginas donde se describirán servicios que ofrece la AFA a las familias, como por ejemplo el comedor, las tardes de junio etc. En la sección del cuerpo que es lo cambiante se mostraran descripciones textuales, sea por ejemplo el comedor. Donde se usará la heurística de “Conexión entre el sistema y el mundo real”, permitiendo al usuario reconocer algo como los títulos, que se pueden encontrar en el mundo real en los diarios, películas, etc. Así también el uso de la heurística “Diseño práctico y minimalista”, así como la heurística de “Reconocer mejor que recordar”, porque en ciertos apartados se tendrá la capacidad de hacer click y mostrar galerías de imágenes pertenecientes al apartado seleccionado.

PÁGINAS DESCRIPCIÓN DE SERVICIOS



4. Desarrollo

4.1 Tecnologías utilizadas

Como en toda implementación a parte de la decisión de la metodología a usar y los patrones de diseño a usar, así como el análisis de requerimientos y el análisis heurístico de los diseños, también se ha de analizar las tecnologías a usar durante el desarrollo. Que tecnologías son las más idóneas para el desarrollo y o tipo de desarrollo a enfrentar.

Para nuestro caso sabemos que la aplicación es una herramienta web que está destinada para las familias en general de la escuela, así como también para las personas que se encuentren bajo el estatus de administradores de la AFA o asociación de familias para la cual se está desarrollando nuestra aplicación.

Por otra parte, se sabe que ha de ser una aplicación web amigable, rápida, eficaz a la hora de presentar contenido. También conocemos que, como organismo no gubernamental ni empresa privada, sino que el AFA es una organización sin lucro y los presupuestos para sus fines son limitados, debemos sopesar que no podrá existir durante el desarrollo tecnologías que tengan un costo elevado o particularmente no factible. Es así como cada tecnología para nuestro caso no solo es evaluada desde el punto de vista tecnológico sino también económico.

Es por todo lo mencionado líneas arriba que se ha decidido el uso de las siguientes tecnologías que mencionare y a las cuales se le hará una breve definición.

4.1.1 BootStrap

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.

Bootstrap es usado por la NASA y la MSNBC, entre otras organizaciones.

Bootstrap, originalmente llamado Blueprint de Twitter, fue desarrollado por Mark Otto y Jacob Thornton de Twitter, como un marco de trabajo (framework) para fomentar la consistencia entre las herramientas internas. Antes de Bootstrap, se usaron varias bibliotecas para el desarrollo de interfaces de usuario, lo que generó inconsistencias y una gran carga de trabajo en su mantenimiento.

En agosto del 2011, Twitter liberó Bootstrap como código abierto. En febrero del 2012, se convirtió en el proyecto de desarrollo más popular de GitHub.

Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo LESS que implementan la variedad de componentes de la herramienta. Una hoja de estilo llamada bootstrap.less incluye los componentes de las hojas de estilo. Los desarrolladores pueden

adaptar el mismo archivo de Bootstrap, seleccionando los componentes que deseen usar en su proyecto.

Los ajustes son posibles en una medida limitada a través de una hoja de estilo de configuración central. Los cambios más profundos son posibles mediante las declaraciones LESS.

El uso del lenguaje de hojas de estilo LESS permite el uso de variables, funciones y operadores, selectores anidados, así como clases mixin.

Después de la definición que se ha introducido sobre Bootstrap, esta tecnología es seleccionada por el beneficio que permite no solo en la visualización del diseño y por el uso de CSS 4, así también porque permite la creación de diseños responsive usando la gran variedad de clases que nos otorga.

4.1.2 jQuery

jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.¹ Fue presentada el 14 de enero de 2006 en el BarCamp NYC. De acuerdo con un análisis de la Web (realizado en 2017) JQuery es la biblioteca de JavaScript más utilizada, por un amplio margen.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados.⁴ JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

La sintaxis de JQuery está diseñada para facilitar la navegación por un documento, seleccionar elementos DOM, crear animaciones, manejar eventos y desarrollar aplicaciones AJAX. JQuery también proporciona capacidades para que los desarrolladores creen complementos en la parte superior de la biblioteca de JavaScript. El enfoque modular de la biblioteca JQuery permite la creación de poderosas páginas web dinámicas y aplicaciones web.

El conjunto de características principales de JQuery (selecciones de elementos DOM, recorrido y manipulación) habilitados por su motor de selección (denominado "Sizzle" de v1.3), crearon un nuevo "estilo de programación", algoritmos de fusión y estructuras de datos DOM. Este estilo influyó en la arquitectura de otros marcos de JavaScript como YUI v3 y Dojo, lo que más tarde estimuló la creación de las API de selectores estándar.

jQuery, en su núcleo, es una biblioteca de manipulación de Modelo de Objetos del Documento (DOM). El DOM es una representación en estructura de árbol de todos los elementos de una página web. JQuery simplifica la sintaxis para buscar, seleccionar y manipular estos elementos DOM. Por ejemplo, JQuery se puede usar para encontrar un elemento en el documento con una determinada propiedad (por ejemplo, todos los elementos con una etiqueta h1), cambiar uno o más de sus atributos (por ejemplo, color, visibilidad) o hacer que responda a un evento (por ejemplo, un clic del ratón).

jQuery también proporciona un paradigma para el manejo de eventos que va más allá de la selección y manipulación básicas de elementos DOM. La asignación de eventos y la definición de la función de devolución de llamada de eventos se realizan en un solo paso en una única ubicación en el código. JQuery también pretende incorporar otra funcionalidad de JavaScript muy utilizada (por ejemplo, entradas y salidas de fundido cuando se ocultan elementos, animaciones mediante la manipulación de las propiedades de CSS).

Después de la definición sobre JQUERY, esta tecnología no será usada directamente dentro de nuestra aplicación, pero si es usada dentro de Bootstrap bajo la cual podemos hacer uso para determinados controles como puede ser un acordeón.

4.1.3 React

React (también llamada React.js o ReactJS) es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Es mantenido por Facebook y la comunidad de software libre. En el proyecto hay más de mil desarrolladores libres.

React intenta ayudar a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo. Su objetivo es ser sencillo, declarativo y fácil de combinar. React sólo maneja la interfaz de usuario en una aplicación; React es la Vista en un contexto en el que se use el patrón MVC (Modelo-Vista-Controlador) o MVVM (Modelo-vista-modelo de vista). También puede ser utilizado con las extensiones de React-based que se encargan de las partes no-UI (que no forman parte de la interfaz de usuario) de una aplicación web.

React fue creado por Jordan Walke, un ingeniero de software de Facebook, quien liberó un primer prototipo de React llamado "FaxJS". Este fue influenciado por XHP de HTML una librería de componentes para PHP. Este fue usado por primera vez en el Feed de Noticias de Facebook en 2011 y después en Instagram en 2012.

Se volvió código abierto en la JSConf US in May 2013. React Native, permitió el desarrollo nativo para Android, iOS y UWP con React, esto fue anunciado por Facebook en la React Conf en febrero de 2015 y se volvió código abierto en marzo de 2015.

El 18 de Abril de 2017, Facebook anuncio React Fiber, un nuevo conjunto de algoritmos internos para la renderización, esto es opuesto al viejo algoritmos de renderizado de React.

Fiber divide la animación en segmentos que se pueden distribuir en varios fotogramas. Asimismo, la estructura de una página se puede dividir en segmentos que se pueden mantenerse y actualizar por separado. Las funciones de JavaScript y el objeto del DOM virtual son llamados "fibers" y cada uno se puede operar y actualizar por separado, lo que permite una representación en pantalla más suave React mantiene un virtual DOM propio, en lugar de confiar solamente en el DOM del navegador. Esto deja a la biblioteca determinar qué partes del DOM han cambiado comparando contenidos entre la versión nueva y la almacenada en el virtual DOM, y utilizando el resultado para determinar cómo actualizar eficientemente el DOM del navegador. Propiedades (props) de react.

Después de la definición que se ha introducido sobre REACT, esta tecnología es electa no solo por ser una de las nuevas tecnologías en el desarrollo del front de una aplicación, sino también porque gracias a su desarrollo en javascript permite el uso de una tecnología usada con anterioridad como es javascript para la creación de métodos, clases, etc. Así también react nos brinda esa rapidez en la ejecución de la aplicación que con otros frameworks debiéramos controlar usando quizás mas cache por parte del servidor o ampliar quizás la

memoria del servidor a usar de acuerdo con la carga de imágenes, código no interpretado, etc.

4.1.4 Axios

Axios es una librería JavaScript que puede ejecutarse en el navegador y que nos permite hacer sencillas las operaciones como cliente HTTP, por lo que podremos configurar y realizar solicitudes a un servidor y recibiremos respuestas fáciles de procesar.

El uso de axios es electo porque, así como se desarrolla una parte front en toda aplicación, también en algunos casos se ha de pensar en el uso de tecnologías back, sobre las cuales podemos obtener datos o funcionalidades, sean estas propias o externas. Entonces la mejor comunicación que se puede realizar es el realizar una petición http y esperar la respuesta, para todo esto Axios nos es de gran ayuda.

4.1.5 Firebase

Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles lanzada en 2011 y adquirida por Google en 2014. Es una plataforma ubicada en la nube, integrada con Google Cloud Platform, que usa un conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad, haciendo posible el crecimiento del número de usuarios y dando resultado también a la obtención de una mayor monetización.

Ventajas que se adquieren al desarrollar un proyecto con Firebase:

Sincronizar fácilmente los datos de los proyectos sin tener que administrar conexiones o escribir lógica de sincronización compleja.

Uso de un conjunto de herramientas multiplataforma, se integra fácilmente para plataformas web como en aplicaciones móviles. Es compatible con grandes plataformas, como IOS, Android, aplicaciones web, Unity y C++.

Uso de la infraestructura de Google y escala automática para cualquier tipo de aplicación, desde las más pequeñas hasta las más potentes.

Creación de proyectos sin necesidad de un servidor, las herramientas se incluyen en los SDK para los dispositivos móviles y web, por lo que no es necesario la creación de un servidor para el proyecto.

Firebase dota a sus usuarios de una gran documentación para crear aplicaciones usando esta plataforma. Aparte de esto, ofrece soporte gratuito mediante correo electrónico para todos sus usuarios, y además sus desarrolladores participan activamente en plataformas como Github y StackOverflow, así como poseen un canal de YouTube explicando el funcionamiento de varias de sus herramientas.

Gracias a todas estas funcionalidades, como desarrolladores podemos combinar y adaptar la plataforma a la medida de nuestras necesidades.

La elección de esta tecnología viene aplicada por el factor económico y tecnológico, como bien hemos hablado se requiere poder desarrollar una aplicación web que sea rápida, ligera y segura. Firebase nos brinda todo esto, pues firebase no solo contiene una BBDD, sino

también muchas otras funcionalidades que hemos definido y que permitirán su uso sin coste relevante para una aplicación web que tendrá poco almacenamiento y media carga de solicitudes. Todo esto permite no realizar pago alguno, pues para el uso y carga de las imágenes no se sobrepasarán los límites gratuitos en todo momento.

4.1.6 Fontawesome

Fontawesome es una librería de iconos y tipologías de letras, aunque para nuestro caso la usaremos por los iconos que contiene para poder brindar una mayor visibilidad en menús y ciertos apartados dentro de la aplicación web.

4.1.7 Tinymce

TinyMCE es un editor de texto WYSIWYG para HTML de código abierto que funciona completamente en JavaScript y se distribuye gratuitamente bajo licencia LGPL. Al ser basado en JavaScript TinyMCE es independiente de la plataforma y se ejecuta en el navegador de internet.

Esta herramienta nos permitirá darle al usuario administrador la capacidad de poder formatear de la manera que elija los textos y/o descripciones que se agreguen en todas las páginas correspondientes a nuestra aplicación sobre las cuales el administrador pueda editar o modificar el cuerpo del texto.

4.2 Arquitectura de la aplicación

En este apartado definiremos la arquitectura de nuestro desarrollo, como se ha implementado los datos y la aplicación web.

4.2.1 MVC

MVC (Modelo-Vista-Controlador) es un patrón en el diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Enfatiza una separación entre la lógica de negocios y su visualización.

En la aplicación hemos usado el modelo anteriormente definido, porque con ello podemos permitir un desarrollo más limpio y estructurado.

Este patrón nos permite organizar efectivamente los ficheros, además de permitir un seguimiento más legible del desarrollo de la aplicación.

La ventaja del uso de este patrón es poder tener separado el código y poder identificar así que parte del código se encarga en cada instante durante la ejecución de cada proceso relativo a la misma.

Por otra parte, se harán las separaciones respectivas dentro de la aplicación para que solo la vista se use para presentar los datos, el controlador para realizar las operaciones pertinentes antes de ser presentados los datos y la parte del modelo que contendrá las definiciones de los objetos que se usarán respectivamente dentro de la aplicación.

4.2.2 Api Rest

Las API REST proporcionan una forma flexible y ligera de integrar aplicaciones y han surgido como el método más común para conectar componentes en la arquitectura de microservicios

Nos permite mantener separada en un principio todo aquello que ataque directamente a una BBDD, así como implementar servicios que deban interactuar con otros servicios y/o aplicaciones que al final brinden un tipo de información que será consumida por una aplicación que modele y presente la información de acuerdo con las normas y características de cada aplicación diseñada para brindar la información a los usuarios finales.

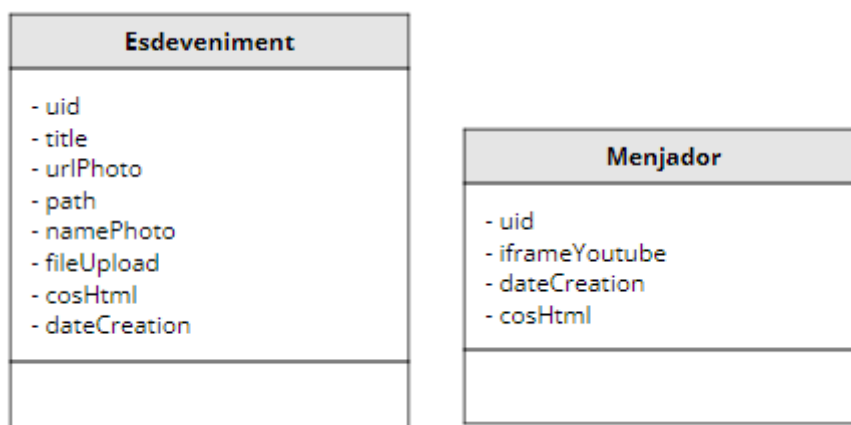
En el caso de nuestra aplicación se hará uso de la generación de un api rest para poder brindar servicios en un principio de mail pues en una de las tecnologías usadas, aunque permite el uso de llamadas a determinadas aplicaciones fuera de nuestra aplicación, esto permitiría revelar claves sensibles del proyecto, por ello se ha decidido la generación del api para poder usarla y así no tener que descubrir datos sensibles respecto a nuestra aplicación. También estará preparada por si se realiza algún cambio que necesite la generación de nuevos end points para resolver cualquier otra característica o se requiera dar solución a algún proceso que requiera su separación.

4.2.3 Base de datos

En la aplicación se decidió que usaríamos Firebase, no solo por motivo económico debido a la cantidad de datos, también porque al contener una BBDD no relacional como si hablásemos de mongo DB nos permitirá un acceso más rápido a la información.

Nuestra base de datos almacenara n cantidad de documentos de tipos distintos, que no se relacionaran mediante un ID, sin embargo, sí que contendrán in ID pero de manera identificativa para ser usado en cuanto pedir dicho documento sea como un ente único o como perteneciente a una colección.

Documentos que almacenaremos en nuestra BBDD no relacional:

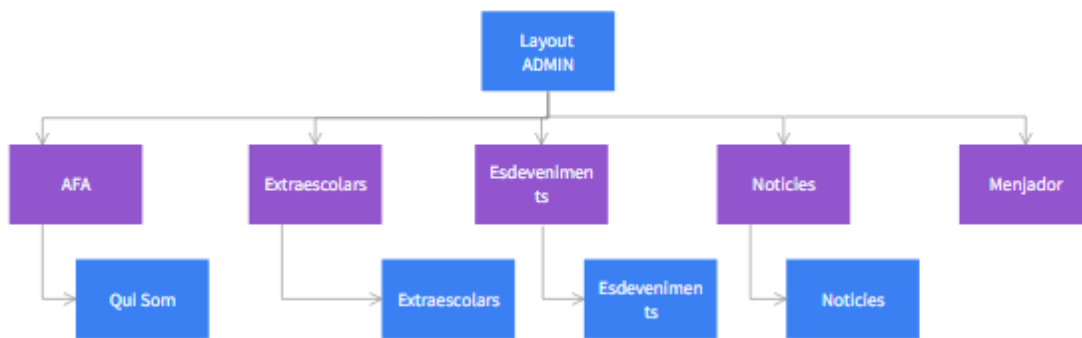


Extraescolar
<ul style="list-style-type: none"> - uid - title - whenDO - urlPhoto - price - parragraph - namePhoto - mainText - howTimes - grupsToDo - dateEnd - dateStart

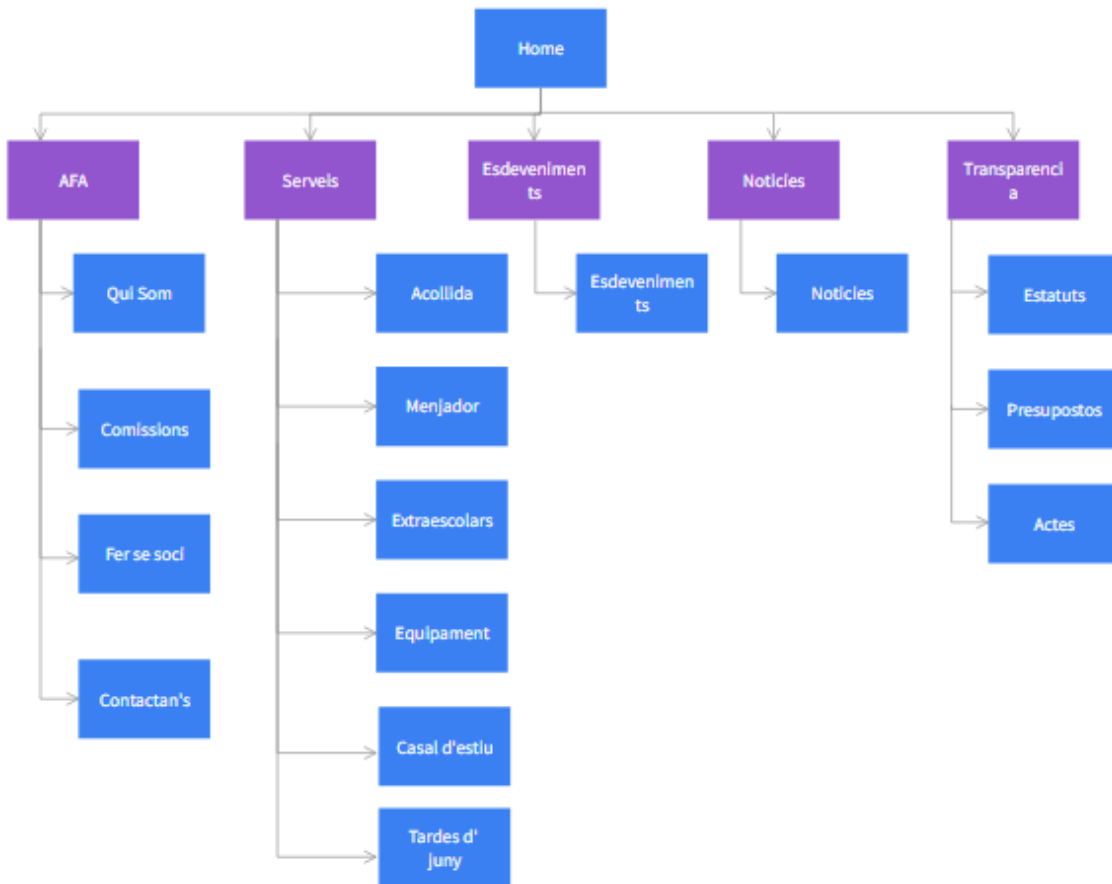
Noticie
<ul style="list-style-type: none"> - uid - title - urlPhoto - pathslmages - namePhoto - path - fileUpload - dateCreation - cosHtml

4.2.4 Esquema web

Esquema web pages admin aplicaci3n



Esquema web pages aplicaci3n



4.2.5 Aplicando patr3n de arquitectura MVC

A continuaci3n, en este apartado se desarrollar3 como se implanto MVC en este TFG, usando para ello las definiciones de los modelos, vistas y controladores creados para cada uno de los fines mencionados.

4.2.5.1 Modelos

Los modelos son 3tiles dentro de la aplicaci3n tanto a nivel front o back de la aplicaci3n, para poder ser el objeto que contiene la informaci3n intercambiada y as3 prestar ayuda a la hora del intercambio de datos que transcurre durante la ejecuci3n de cualquier funcionalidad dentro de la aplicaci3n.

Modelos generados dentro de la aplicaci3n:

Extraescolars:

```
export class extraEscolars{
  constructor(Uid,path,plazas,title,parragraph,dateStart,dateEnd,mainText,namePhoto,urlPhoto,whenDo,howTimes,price,grupsToDo){
    this.Uid = Uid;
    this.path = path;
    this.plazas = plazas;
```

```

    this.title= title;
    this.parragraph= parragraph;
    this.dateStart= dateStart;
    this.dateEnd= dateEnd;
    this.mainText= mainText;
    this.namePhoto= namePhoto;
    this.urlPhoto= urlPhoto;
    this.whenDo= whenDo;
    this.howTimes= howTimes;
    this.price= price;
    this.grupsToDo= grupsToDo;
  }
}

```

Esdeveniment:

```

export class esdeveniment{
  constructor(uid,path,fileupload,title,cosHtml,dateCreation,namePhoto,urlPhoto){
    this.uid = uid;
    this.path = path;
    this.fileupload = fileupload;
    this.title = title;
    this.cosHtml = cosHtml;
    this.dateCreation = dateCreation;
    this.namePhoto= namePhoto;
    this.urlPhoto= urlPhoto;
  }
}

```

Noticie:

```

export class noticia{
  constructor(uid,path,fileupload,title,cosHtml,dateCreation,namePhoto,urlPhoto, pathsImages,
    urlsPathslImages,ImagesGalery){
    this.uid = uid;
    this.path = path;
    this.fileupload = fileupload;
    this.title = title;
    this.cosHtml = cosHtml;
    this.dateCreation = dateCreation;
    this.namePhoto= namePhoto;
    this.urlPhoto= urlPhoto;
    this.pathsImages = pathsImages;
    this.urlsPathslImages = urlsPathslImages;
    this.ImagesGalery = ImagesGalery;
  }
}

```

Mail:

```

export class mail {
  constructor(from,to,subject,of,text, fechaCreacion){
    this.from = from;
  }
}

```



```

    this.to = to;
    this.subject = subject;
    this.of= of;
    this.text = text;
    this.fechaCreacion = fechaCreacion;
  }
}

```

Comission:

```

export class comissio{
  constructor(uid,path,fileupload,title,cosHtml,dateCreation,namePhoto,urlPhoto,listaUsuarios){
    this.uid = uid;
    this.path = path;
    this.fileupload = fileupload;
    this.title = title;
    this.cosHtml = cosHtml;
    this.dateCreation = dateCreation;
    this.namePhoto= namePhoto;
    this.urlPhoto= urlPhoto;
    this.listaUsuarios= listaUsuarios;
  }
}

```

InscripcionComission:

```

export class inscripcionComission{
  constructor(nombre,primerApellido,segundoApellido,email,fechaNacimiento,fechaInscripcion,comision){
    this.nombre = nombre;
    this.primerApellido = primerApellido;
    this.segundoApellido = segundoApellido;
    this.email = email;
    this.fechaNacimiento = fechaNacimiento;
    this.fechaInscripcion = fechaInscripcion;
    this.comision = comision;
  }
}

```

Menjador:

```

export class menjador{
  constructor(uid,cosHtml,dateCreation,iframeYoutube){
    this.uid = uid;
    this.cosHtml = cosHtml;
    this.dateCreation = dateCreation;
    this.iframeYoutube= iframeYoutube;
  }
}

```

modelViewRegistroAlumno:

```

export class modelViewRegistroAlumno{
  constructor(uid,titleExtraEscolar,nombreTutor,apellidosTutor,email,nombreAlumno,apellidosAlumno,edad,curso,grupos){

```

```

    this.uid = uid;
    this.titleExtraEscolar= titleExtraEscolar;
    this.nombreTutor= nombreTutor;
    this.apellidosTutor = apellidosTutor;
    this.email= email;
    this.nombreAlumno= nombreAlumno;
    this.apellidosAlumno = apellidosAlumno;
    this.edad = edad;
    this.curso= curso;
    this.grupos = grupos;
}
}

```

Estatut:

```

export class estatut{
    constructor(uid,bodyHtml,namesPDFS,urlsPDFS,pathsPDFS){
        this.uid = uid;
        this.bodyHtml = bodyHtml;
        this.namesPDFS = namesPDFS;
        this.urlsPDFS = urlsPDFS;
        this.pathsPDFS = pathsPDFS;
    }
}

```

Actes:

```

export class actes{
    constructor(uid,bodyHtml,namesPDFS,urlsPDFS,pathsPDFS){
        this.uid = uid;
        this.bodyHtml = bodyHtml;
        this.namesPDFS = namesPDFS;
        this.urlsPDFS = urlsPDFS;
        this.pathsPDFS = pathsPDFS;
    }
}

```

Acollida:

```

export class acollida{
    constructor(uid,cosHtml,dateCreation){
        this.uid = uid;
        this.cosHtml = cosHtml;
        this.dateCreation = dateCreation;
    }
}

```

CasalEstiu:

```

export class casalEstiu{
    constructor(uid,cosHtml,dateCreation,urlPhoto,path,namePhoto){
        this.uid = uid;

```

```
    this.cosHtml = cosHtml;
    this.dateCreation = dateCreation;
    this.urlPhoto = urlPhoto;
    this.path = path;
    this.namePhoto = namePhoto;
  }
}
```

TardesJuny:

```
export class tardesJuny{
  constructor(uid,cosHtml,dateCreation,urlPhoto,path,namePhoto){
    this.uid = uid;
    this.cosHtml = cosHtml;
    this.dateCreation = dateCreation;
    this.urlPhoto = urlPhoto;
    this.path = path;
    this.namePhoto = namePhoto;
  }
}
```

Equipament:

```
export class equipament{
  constructor(uid,cosHtml,dateCreation,urlPhoto,path,namePhoto){
    this.uid = uid;
    this.cosHtml = cosHtml;
    this.dateCreation = dateCreation;
    this.urlPhoto = urlPhoto;
    this.path = path;
    this.namePhoto = namePhoto;
  }
}
```

Quisom:

```
export class quisom{
  constructor(uid,cosHtml,dateCreation,urlPhoto,path,namePhoto){
    this.uid = uid;
    this.cosHtml = cosHtml;
    this.dateCreation = dateCreation;
    this.urlPhoto = urlPhoto;
    this.path = path;
    this.namePhoto = namePhoto;
  }
}
```

FesteSoci:

```
export class festeSoci{
  constructor(uid,cosHtml,dateCreation,urlPhoto,path,namePhoto){
    this.uid = uid;
    this.cosHtml = cosHtml;
    this.dateCreation = dateCreation;
```

```

this.urlPhoto = urlPhoto;
this.path = path;
this.namePhoto = namePhoto;
}
}

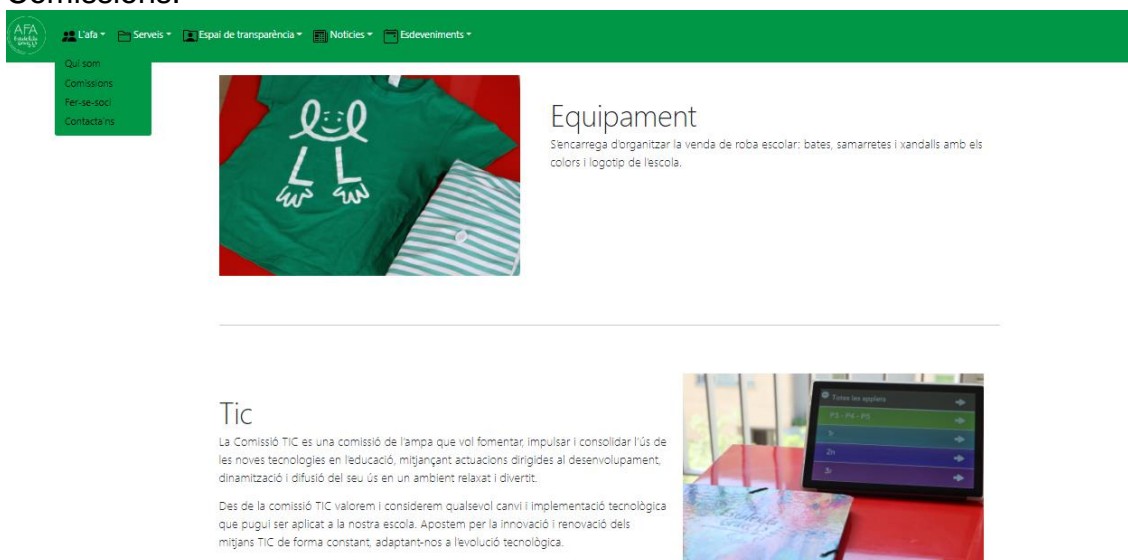
```

4.2.5.2 Vistas

En este apartado se mencionarán las distintas vistas que se usarán a lo largo del desarrollo de la aplicación. Así como también los componentes que las componen. React recordemos que son componentes que se renderizan en la aplicación para la creación de vistas.

A continuación, las vistas y componentes creados para la aplicación.

Comissions:



Esta página dentro de su diseño contiene dos componentes a parte de un componente formulario para el desarrollo de la misma.

Componente RectangleCard.

```

function RectangleCard({Id,path,title,urlImage,arrayData, classnameBody, classnaImage, handleRemove, handleEdit,
componentcall}) {
  const buttonsAdmin = (componentcall) => {
    if (componentcall === "admin") {
      return(
        <div className="row g-0">
          <div className="col-6">
            <a href="#ass" name={Id} onClick={handleEdit} className="nav-link px-1" role="button" aria-
disabled="true">Editar</a>
          </div>
          <div className="col-6">
            <a href="#ass" name={` ${Id} - ${path}`} onClick={handleRemove} className="nav-link px-1" role="button"
aria-disabled="true">Eliminar</a>
          </div>
        </div>
      )
    }
  }
}

```

```

    }else{
      return(
        <></>
      )
    }
  }
}

const drawRectangle = (componentcall) => {
  if (componentcall === "admin") {
    return(
      <>
      <hr className="featurette-divider"></hr>
      <div className="row featurette">
        <div className={classnameBody}>
          {
            buttonsAdmin(componentcall)
          }
          <h2 className="featurette-heading"><span className="text-muted">{title}</span></h2>
          <p className="lead">{arrayData}</p>
        </div>
        <div className={classnameImage}>
          <img className="bd-placeholder-img bd-placeholder-img-lg featurette-image img-fluid mx-auto"
width="500" height="500" src={urlImage} alt=""/>
        </div>
      </div>
    </>
  )
}else{
  return(
    <>
    <hr className="featurette-divider"></hr>
    <a href={` /galleria/${Id} `}>
      <div className="row featurette">
        <div className={classnameBody}>
          {
            buttonsAdmin(componentcall)
          }
          <h2 className="featurette-heading"><span className="text-muted">{title}</span></h2>
          <p className="lead">{arrayData}</p>
        </div>
        <div className={classnameImage}>
          <img className="bd-placeholder-img bd-placeholder-img-lg featurette-image img-fluid mx-auto"
width="500" height="500" src={urlImage} alt=""/>
        </div>
      </div>
    </a>
    </>
  )
}
}
}

```

```

return (
  <>
    {
      drawRectangle(componentcall)
    }
  </>
)
}
export default RectangleCard

```

Componente ListRectangleCard:

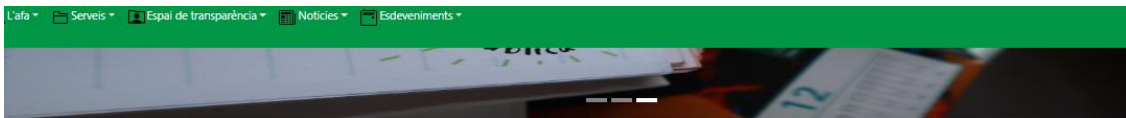
```

import RectangleCard from "./RectangleCard";

function ListRectangleCard({arrayData, handleEdit, handleRemove, componentCall, nameList}) {
  if (!arrayData.length) return <h3>No existe ningun registro</h3>;
  return (
    <>
      <hr className="featurette-divider"></hr>
      <div className="container">
        <h1>{nameList}</h1>
        <div className="container mb-3">
          <div className="row">
            {
              arrayData.map((esdevenim,index) => (
                <RectangleCard Id={esdevenim.uid} arrayData={esdevenim.cosHtml} path={esdevenim.path}
                classnameBody={index%2 === 0 ? "col-md-7":"col-md-7 order-md-2"} classnamelImage={index%2 === 0 ? "col-md-5":"col-
                md-5 order-md-1"} handleEdit={handleEdit} key={esdevenim.uid} title={esdevenim.title} urlImage={esdevenim.urlPhoto}
                handleRemove={handleRemove} componentcall={componentCall}/>
              ))
            }
          </div>
        </div>
      </div>
    </>
  )
}
export default ListRectangleCard

```

Contactans:



Contacta amb nosaltres.

Si tens algun dubte, consulta o necessites enviar-nos alguna informació no dubtis a fer servir aquest formulari per contactar amb nosaltres.

Internamente esta vista está compuesta por un componente denominado

RectangleWithForm:

```
function RectangleWithForm({handleSubmit, handleChange}) {
  return (
    <>
      <hr className="featurette-divider"></hr>
      <div className="row featurette">
        <div className="col-md-7">
          <h2 className="featurette-heading">Contacta amb nosaltres.</h2>
          <p className="lead">Si tens algun dubte, consulta o necessites enviar-nos alguna informació no dubtis a fer servir aquest formulari per contactar amb nosaltres.</p>
        </div>
        <div className="col-md-5">
          <div className="card border-info mb-3">
            <div className="card-header bg-warning"><h2 className="card-title">Contacta'ns</h2></div>
            <div className="card-body">
              <form onSubmit={handleSubmit}>
                <div className="mb-3">
                  <label for="inputNombre" className="form-label">Nom i Cognoms</label>
                  <input type="text" onChange={handleChange} className="form-control" id="inputNombre" name="nombre"/>
                </div>
                <div className="mb-3">
                  <label for="inputSubject" className="form-label">Asunto</label>
                  <input type="text" onChange={handleChange} className="form-control" id="inputSubject" name="subject"/>
                </div>
                <div className="mb-3">
                  <label for="inputEmail" class="form-label">Correu electrònic</label>
                  <input type="email" onChange={handleChange} className="form-control" id="inputEmail" aria-describedby="emailHelp" name="email"/>
                </div>
              </form>
            </div>
          </div>
        </div>
      </div>
    </>
  );
}
```

```

        <div className="mb-3">
          <label for="formControlTextarea1" className="form-label">Missatge</label>
          <textarea className="form-control" onChange={handleChange} id="formControlTextarea1" rows="3"
name="textArea"></textarea>
        </div>
        <div className="d-grid gap-2">
          <button type="submit" className="btn btn-primary">Enviar</button>
        </div>
      </form>
    </div>
  </div>
</div>
</div>
</div>
)
}
export default RectangleWithForm

```

FerseSoci:

The screenshot shows the website for 'Fes te soci' (Become a member). The header is green with the AFA logo and navigation links: 'L'afa', 'Serveis', 'Espai de transparència', 'Notícies', and 'Esdeveniments'. The main heading is 'Fes te soci'. Below it, there is a horizontal line and a paragraph of text explaining the benefits of being a member. A second horizontal line follows. Below that, there are three input fields for registration: 'Email', 'Nombres', and 'Primer Apellido'.

Componente CardHorizontalWithForm

```

function CardHorizontalWithForm({EnviarMail}) {
  return (
    <>
      <hr className="featurette-divider"></hr>
      <div className="containerH1"><h1 className="text-h1">Fes te soci</h1></div>
      <div className="container">
        <hr className="featurette-divider"></hr>
        <p className="lead">Les famílies sòcies de l'AFA gaudeixen de descomptes en el preu dels serveis d'acollida, menjador i extraescolars, entre d'altres avantatges. La quota d'inscripció permet mantenir aquests serveis, així com altres activitats i equipaments de l'escola.</p>
        <p className="lead">*PER FORMAR PART DE L'AFA CAL SER MARE, PARE O TUTOR/A LEGAL D'UN INFANT DE L'ESCOLA.</p>
        <hr className="featurette-divider"></hr>
        <form onSubmit={EnviarMail}>

```



```

<div className="mb-3">
  <label for="exampleInputEmail1" className="form-label">Email</label>
  <input type="email" className="form-control" id="exampleInputEmail1" aria-describedby="emailHelp"/>
</div>
<div className="mb-3">
  <label for="exampleInputPassword1" className="form-label">Nombres</label>
  <input type="text" className="form-control" id="exampleInputPassword1"/>
</div>
<div className="mb-3">
  <label for="exampleInputPassword1" className="form-label">Primer Apellido</label>
  <input type="text" className="form-control" id="exampleInputPassword1"/>
</div>
<div className="mb-3">
  <label for="exampleInputPassword1" className="form-label">Segundo Apellido</label>
  <input type="text" className="form-control" id="exampleInputPassword1"/>
</div>
<div className="mb-3">
  <label for="exampleInputPassword1" className="form-label">DNI/NIE</label>
  <input type="text" className="form-control" id="exampleInputPassword1"/>
</div>
<div className="mb-3">
  <label for="exampleInputPassword1" className="form-label">Fecha Nacimiento</label>
  <input type="date" className="form-control" id="exampleInputPassword1"/>
</div>
<button type="submit" className="btn btn-primary">Aceptar</button>
</form>
</div>
<hr className="featurette-divider"></hr>
</>
)
}
export default CardHorizontalWithForm

```

4.2.5.3 Controlador

En este apartado se explica la funcionalidad de cada controlador, los controladores que usamos para que la ejecución de las funcionalidades internas dentro de la aplicación.

Cabe señalar que al hacer uso de Firebase nos ahorramos gran cantidad de código para la seguridad de la aplicación como es el caso del login de usuario. Por lo que cabe mencionar en este apartado como trabaja firebase.

4.2.5.3.1 Firebase

Es más que una BBDD, es una plataforma, la cual nos brinda muchas características, entre ella la de almacenar datos. Estos datos no son relacionales. Es más semejante a Mongo DB una BBDD no relacional. Por lo cual lo que almacenamos en ella no son datos, sino objetos. Objetos json que contienen los datos, como otra característica nos brinda un ID para cada dato almacenado con lo cual no es necesario la creación de una identidad para ser enviado cuando se almacena cada objeto, esto lo hace por nosotros firebase.

Los datos almacenados tienen distintos tipos de seguridad. Esta seguridad es brindada por scripts, scripts en los cuales puedes brindar seguridad a cada tipo de almacenamiento dentro de firebase. Por ejemplo, puedes brindar seguridad por paths.

A algunos paths quizás interesaría solo la seguridad fuera por autenticación, vale decir que solo los usuarios autenticados tendrán acceso al recurso, pero a su vez podrías indicar que solo fuera el acceso de lectura y no de escritura o viceversa. O quizás resulte mejor indicar que a dicho recurso solo podrán acceder para lectura y escritura los usuarios que sean autenticados y que sean los dueños de los datos almacenados. Cada entidad u objeto almacenado tiene referencia al id de quien lo almacena, esto no es visible directamente en el objeto, pero si cuando uno lo recupera esta información es indicada, de esta manera se puede comprobar la propiedad del usuario. Esto a su vez es una característica de la seguridad y puede ser indicada en los scripts para que cuando un usuario almacene o recupere alguna información pueda ser verificada y así pueda proceder la petición que haya indicado.

Por otro lado, al poder definir la seguridad de los datos por script y por paths, esto indica que se pueden excluir ciertos datos de la seguridad o brindar cualquier otro tipo de seguridad a cada uno de los distintos paths donde se almacena la información. Esto no solo ocurre con firestore, sino también con firestorage. Ambos son parte de las características de firebase para el almacenamiento.

Existen la posibilidad de asegurar un path de información por el criterio de que el usuario sea administrador únicamente para así poder realizar este tipo de petición o podríamos indicar que quien use el recurso sea un usuario que sea propietario, que lo haya almacenado, que sea administrador o que tenga algún tipo de permiso específico y que pueda acceder a otro path específico como parte de la exclusión.

4.2.5.3.1.1 FileStorage

Permite al usuario almacenar imágenes o ficheros, esto ficheros al igual que con firestore pueden tener seguridad, seguridad que se desarrollara a partir del script creado para el path sobre el cual se indique su aplicación. Digamos que tenemos una carpeta llamada imágenes y otra que se llame ficheros/pdfs y ambos sean del mismo propietario, pero que el usuario solo pueda agregar o ver lo existente y dichos paths, sin embargo, no pueda eliminar. Quien lo eliminaría solo podría ser un usuario que, aunque no es dueño de los ficheros es administrador de la plataforma y por ello se le concede dicho acceso a la par de la escritura y la lectura.

Auth es otra de las características que nos permite firebase, esta característica nos brinda muchos tipos de acceso.

4.2.5.3.1.2 Authentication

En la authentication hemos elegido el uso de mail and password, esto nos permite crear usuarios desde un email y password para después poder verificarlo y dar el acceso correspondiente. Firebase nos proporciona un SDK, este kit de herramientas de desarrollo nos permite desarrollar con más facilidad, pues nos brinda gran cantidad de características para desarrollar funcionalidades que normalmente podría tomar su desarrollo días o semanas, o deber tener un gran grupo de desarrollo para poder así subdividir de tal manera las tareas que así se llegue a un resultado aproximado a lo que nos brinda Firebase. Como se ha indicado nos facilita la creación de usuario seleccionado la opción deseada de como

validaremos estos usuarios. Es por ello que Firebase así como nos ha brindado la opción de poder ingresar y validar un usuario por mail and password o su creación, también existen otros tipos o maneras las cuales solo mencionare a continuación.

- Telefono
- Google
- Facebook
- Twitter
- Apple
- GitHub
- Microsoft
- Yahoo
- Anonimo

En cada uno de los tipos mencionados anteriormente, se ha de tener un usuario valido para estas aplicaciones. Por ejemplo, para Google, la cuenta ha de ser Google. Ha de ser un email de Google para poder acceder y en su acceso se presentará el login de Google para poder realizar el acceso correspondiente.

Así igual pasará en los demás casos. Todos los usuarios de dichas aplicaciones deberán ser validados con un login que mantenga un acceso a dichas aplicaciones y así permitir su validación gracias a algún api key que sea usada desde cada proveedor que permita el acceso.

Esta característica dentro de Auth no solo nos permite realizar una creación de usuario y validación de este. Sino que dentro de la validación podemos obtener un token creado por firebase para no tener que crear dicha funcionalidad, pero, aunque esto se nos brinde por parte firebase, existe la posibilidad de poder crear nuestro propio token a partir de la id de renovación que permite realizar dicha funcionalidad con otros generadores de token que podamos agregar a la seguridad que quisiéramos tener a nivel de usuarios.

Para nuestro caso hemos elegido usar el acceso y creación de usuario mediante el uso de Mail and password. Al ser una aplicación para el acceso de unos usuarios determinados para la generación de contenido nuevo y a sabiendas que no todos los usuarios tienen una cuenta de Facebook, Twitter entre otros. Pero también sabemos que todo usuario tiene una cuenta de mail, por ende, la elección del mail and password se ha creído la más oportuna y necesaria.

Al no tener que crear una estructura dentro de la BBDD como en otras como por ejemplo SQL donde tendríamos que crear una tabla con los campos necesarios donde deberíamos tener un nombre de usuario, mail, password cifrado, datos relativos al usuario como podría ser DNI, Teléfono, etc. todo esto deberíamos de relacionarlo con otra tabla denominada tokens donde se almacenaría el token y la renovación del mismo para poder hacer la validación y generación desde cuando el usuario es agregado a dicha tabla y cuando se otorga el acceso, se tendría que realizar una funcionalidad que permita la renovación del token y que en cada instante este a la espera de las llamadas proporcionadas por el frontend para decidir si el token está a punto de fenecer y así la generación de un token valido para que la sesión del usuario se mantenga hasta que dicho usuario pueda realizar el logout correspondiente.

Una funcionalidad o características adicional que se mantiene dentro de Auth en firebase, es la característica de saber en todo momento el estado del usuario, es decir firebase lo hace por nosotros, nos otorga si el usuario aún mantiene la sesión, si el usuario ha hecho logout, etc. En antiguas aplicaciones o de características más tradicionales se usaban las sesiones, sesiones donde se proporcionaban características del usuario relevantes que permitían generar de alguna manera una clave o key que permitía al usuario mantener la sesión activa, aunque en firebase esto también se puede realizar, ahora bajo las nuevas características y usos de tokens que nos permiten validar el usuario dentro de las peticiones realizadas a las distintas aplicaciones no se hace mucho uso del mismo. Sino que ahora se realiza un llamado a cierta característica que nos permite saber en todo momento si el usuario aun está o no conectado a la aplicación y si aún puede seguir actuando sobre ella, en firebase esto se denomina `onAuthStateChanged` el cual nos permite no solo saber si el usuario este o no logeado, sino su estado en cualquier momento, así como también de donde es el usuario, es decir desde donde llamo a la aplicación, de donde se logeo y desde donde mantiene la sesión. Con lo que el usuario, aunque crea que él puede entrar desde otra ubicación con la cuenta ya validada no podrá salvo realice otro login y con ello una nueva sesión dentro firebase.

Esto permite identificar en cada momento de donde y quien hace uso de la aplicación.

Así Auth de firebase no es solo una herramienta que permita crear usuarios, sino que nos permite hacer uso de ella para todo lo que refiera a un usuario, desde la creación hasta cada estado que el mantenga dentro de nuestra aplicación.

4.2.5.3.2 Controladores dentro de la aplicación

En este apartado se definen y explican los controladores creados dentro de la aplicación para el desarrollo de las funcionalidades dentro de la aplicación.

Controlador CRUDBD, controlador que define e implementa todas las funcionalidades que permiten el manejo y/o envío de los datos a Firebase.

CRUDBD:

```
import { ref, uploadBytes, getDownloadURL, deleteObject } from "firebase/storage";
import { db, imagesRef } from "../firebase/fireabaseConfig";
import { storage } from "../firebase/fireabaseConfig";
import { collection, addDoc, getDocs, onSnapshot, deleteDoc, doc, getDoc, updateDoc } from "firebase/firestore";
export const add = async (tipo, modelo) =>{
  try {
    const docRef = await addDoc(collection(db, tipo), modelo);
    return docRef.id;
  } catch (error) {
    console.log(error);
  }
}
export const getAllCollections = async (tipo) =>{
  const querySnapshot = await getDocs(collection(db, tipo));
  let rst = [];
```

```

querySnapshot.forEach((doc) => {
  let item = doc.data();
  item.id = doc.id;
  rst.push(item);
});
return rst;
}

export const getUrlImage = async (routeRef) =>{
  try {
    const iref = ref(storage,routeRef);
    let urlData = await getDownloadURL(iref);
    return urlData;
  } catch (error) {
    console.log(error);
  }
}

export const removeObject = async (routeRef) =>{
  try {
    const iref = ref(storage,routeRef);
    let objectRemove = await deleteObject(iref);
    return objectRemove;
  } catch (error) {
    console.log(error);
  }
}

export const onGetListTipo = (tipo,callback) => onSnapshot(collection(db,tipo),callback);
export const deleteOneDocOfTipo = async (tipo,uid) => await deleteDoc(doc(db,tipo,uid));
export const getOneDocOfTipo = async (tipo,uid) => await getDoc(doc(db,tipo,uid));
export const updateOneDocOfTpo = async (tipo,uid,modelo) => {
  try {
    const rstUpdate = await updateDoc(doc(db,tipo,uid),modelo);
    console.log(rstUpdate);
  } catch (error) {
    console.log(error);
  }
}

export const uploadFile = async (file,name,uid,userid) => {
  try {
    const iref = ref(imagesRef,`/${userid}/${uid}/${name}`);
    let fileSub = await uploadBytes(iref,file);
    return fileSub;
  } catch (error) {
    console.log(error);
  }
}

```

AUTHCONTEXT: controlador que permite el mantener en todas las funcionalidades los estados necesarios del usuario.

```
import { createContext, useContext, useState, useEffect } from 'react'
import { signInWithEmailAndPassword, createUserWithEmailAndPassword, onAuthStateChanged, signOut } from
"firebase/auth";
import { auth } from "../firebase/fireabaseConfig";

export const authContext = createContext();

export const useAuth = () =>{
  const context = useContext(authContext);
  if (!context) throw new Error("There is no Auth provider");
  return context;
}

export function AuthProvider({children}) {
  const [user,setUser] = useState(null);
  const [loading,setLoading] = useState(true);
  const createUser = async (email, password) => {
    await createUserWithEmailAndPassword(auth, email, password);
  };

  const signUp = async (email,password) =>{
    await signInWithEmailAndPassword(auth,email,password);
  };

  const logOut = async () =>{
    await signOut(auth);
  };

  useEffect(()=>{
    const unsubscribe = onAuthStateChanged(auth, currentUser =>{
      setUser(currentUser);
      setLoading(false);
    })
    return () => unsubscribe();
  })

  return (
    <authContext.Provider value={{signUp, user, createUser, logOut, loading}}>{children}</authContext.Provider>
  )
}
```

Mail: controlador que permite conectarse a nuestro api la cual tiene la funcionalidad de crear mails entre otras. Esta funcionalidad se reaprovechará para la realización de la funcionalidad del mailing para permitir el envío masivo de correspondencia.

```
import axios from "axios";
import { mail } from "../../models/mail";
```

```
export const envioMail = async (user) =>{
  try {
    const data = new mail(user.email,user.email,user.subject,user.nombre,user.textArea);
    axios.post('https://arimathsolutions.com/api/mail',data,{header:{
      'TIPO DE CONTENIDO': 'Aplicación / JSON'
    }})
    .then(res => {
      console.log(res);
      console.log(res.data);
      return true;
    }).catch(error => {
      let errordata = error;
      console.log(errordata);
      return false;
    });
  } catch (error) {
    console.log(error);
    return false;
  }
}
```

5. Conclusiones

En este apartado analizaremos uno a uno los sprint y veremos si se ha podido realizar el trabajo correctamente, Si los objetivos generales de la aplicación han sido totalmente completados.

SPRINT 1:

Historia 1: quisiéramos ver un menú con los apartados necesarios que puedan contener la información que queremos presentar.

Historia 2: quisiéramos poder ver imágenes del AFA de manera que se presenten una a una.

Historia 3: quisiéramos que hubiese otra forma de ingresar a los apartados no solo el menú principal.

En este sprint como objetivo inicial se tenía el menú de la aplicación que permitiera al usuario poder navegar con facilidad dentro de la aplicación, así también el poder mostrar imágenes que tuviesen que ver con la AFA para una llamada de la atención del usuario visitante y una manera alternativa para el llamado de las distintas partes que contiene el sitio web. Después de mencionado el objetivo particular de este sprint y obtener una definición de hecho según cada sprint. Se puede indicar que en dicho sprint se cumplió con el objetivo.

SPRINT 2:

Historia 4: quisiéramos que en el menú AFA hubiese un apartado de Quienes somos.

Historia 5: quisiéramos que en el menú AFA hubiese un apartado de comisiones

Historia 6: quisiéramos que en el menú AFA hubiese un apartado para hacerse socio de la AFA.

En este sprint como objetivo principal según las historias incluidas en el sprint se busca que dentro del menú que contiene la web existan los apartados Quienes somos, comisiones y hacerse socio del AFA. Como indica cada historia, el objetivo principal es que los visitantes puedan saber que es la AFA y como pertenecer a ella. Teniendo una definición de hecho según lo descrito con anterioridad se puede indicar que el objetivo particular del sprint se ha cumplido.

SPRINT 3:

Historia 7: quisiéramos un formulario de contacto.

Historia 8: quisiéramos que en el apartado transparencia exista un menú donde se pueda ver los estatutos de la AFA.

Historia 9: quisiéramos que en el apartado transparencia exista un menú que permita visualizar todos los presupuestos anuales.

En este sprint como objetivo se tiene permitir una comunicación con la AFA mediante el canal de contáctanos, así también el poder brindar información relevante a la transparencia que existe o quiere dar a existir la AFA respecto los presupuestos y estatutos por los que se rigen. Viendo que estas tareas han sido realizadas, podemos indicar que el objetivo particular del sprint se ha alcanzado.

SPRINT 4:

Historia 10: quisiéramos que en el apartado transparencia exista un menú que permita visualizar las actas del año.

Historia 11: quisiéramos que, en el apartado servicios, donde planteamos los servicios que ofrecemos exista un menú que permita ver la información del comedor de la escuela.

Historia 12: quisiéramos que en el apartado servicios también se incluya un menú en el cual se pueda ver la información de la acogida que ofrecemos para la conciliación de la vida laboral de las familias y la escuela de sus hijos.

En este sprint se tiene como objetivo brindar transparencia otorgando así las actas de cada reunión mantenida por la AFA y las familias puedan saber que puntos se trataron para el beneficio de los alumnos de la escuela, así también brindar información que respecta al comedor de la escuela para que las familias sepan que alimentos reciben sus hijos para reforzar su alimentación, como objetivo alterno también el poder ofrecer la información sobre la conciliación entre el trabajo y la familia otorgando así el servicio de acogida en la escuela.

Todo lo mencionado en este sprint se puede dar por cumplido porque el sprint fue realizado y validado.

SPRINT 5:

Historia 13: quisiéramos que dentro del apartado de servicios exista una página donde podamos presentar la información que desarrollamos para las tardes de junio donde la escuela finaliza antes de las 4:30 o 5 para poder conciliar la vida laboral de los tutores legales con la vida escolar de sus hijos.

Historia 14: quisiéramos una página de equipamiento, donde podamos agregar la información relevante del equipamiento anualmente.

Historia 15: quisiéramos una página que contenga la información del casal de verano que se desarrolla tanto en las instalaciones de la escuela como en instalaciones de lugares con los cuales se ha realizado algún convenio para el desarrollo de estas.

En este sprint el objetivo principal ha sido presentar información para dos servicios que desarrollan durante el año curricular la AFA, examinando propuestas y negociando los precios ofertados para que una mayoría de niños puedan acceder, por lo que es necesario la tenencia de un lugar específico donde brindar la información de lo que existirá para dichos servicios, por otro lado también existe el servicio de equipamiento el cual corresponde a los precios del equipamiento con las mejores ofertas que ha podido negociar la AFA como entidad para que los alumnos de la escuela puedan adquirirlos si así lo desearan. Para este Sprint se ha podido alcanzar solo un 70 por ciento del sprint. Pues se ha dejado para un próximo sprint por falta de información lo necesario para el apartado equipamiento.

SPRINT 6:

Historia 16: quisiéramos poder tener una página donde se presenten las extraescolares que planeamos desarrollar durante el año curricular para los alumnos, esto nos parece que debería de ir en el apartado de servicios.

Historia 17: quisiéramos una página que contenga la información de los eventos anuales.

Historia 18: quisiéramos una página que contenga la información de las noticias que comparta la escuela o la AFA a título propio.

En este sprint como objetivo se tiene poder brindar información sobre las extraescolares, eventos y noticias. Las extraescolares al igual que el equipamiento, el comedor, los libros, etc. Es un servicio que ofrece la AFA y se encargan de negociar con distintas empresas sus ofertas y servicios que pueden ofrecer para el desarrollo de una actividad que brinde a los niños no solo una distracción o en que ocupar el tiempo. Sino también que la actividad parta de una expresión educativa. Que consolide o que permita a los niños solventar ciertas habilidades y/o complementar ciertos conocimientos. Por lo desarrollado en este sprint y la aprobación de la AFA, se puede dar por cumplido el objetivo.

SPRINT 7:

Historia 19: quisiéramos poder tener una parte administrativa para en un futuro poder hacer algunas tareas internas en ella.

Historia 20: quisiéramos poder realizar el acceso a dicha área mediante un login usando un mail que todos tenemos, por lo pronto será solo un único usuario.

Historia 21: quisiéramos poder tener dentro del área administrativa un apartado que permita la generación y/o modificación del contenido del apartado comedor.

En este sprint el objetivo ha sido el poder iniciar con la parte administrativa de la web. El lugar que permita hacer ciertas operaciones dentro de la aplicación.

Definido lo anterior se puede dar por cumplido el objetivo al haberse realizado cada tarea del sprint y cumplido con su definición de hecho.

SPRINT 8:

Historia 22: quisiéramos poder tener dentro del área administrativa un apartado que nos permita la generación de eventos que se desarrollan en la escuela.

Historia 23: nos gustaría poder tener un apartado dentro del área administrativa que permita modificar o crear noticias que se desplegaran en la página de noticias.

Historia 24: nos gustaría poder tener dentro del área administrativa un lugar donde poder crear los eventos, editarlos o eliminarlos para que se presentan dentro del apartado de eventos en la página general de la web.

En este sprint como objetivo se envuelve el poder generar contenido, poder modificarlo para que las familias puedan ver el nuevo contenido desplegado en el sitio web. Definido esto se puede dar por cumplido habiendo realizado las tareas del sprint y la obtención de su definición de hecho.

SPRINT 9:

Historia 25: nos sería útil dentro del área administrativa poder agregar comisiones, así como también poder eliminarlas o editarlas.

Historia 26: en el área administrativa nos sería de utilidad poder nosotros agregar, modificar o eliminar las extraescolares que presentaremos cada año curricular.

Historia 27: nos sería de utilidad que desde el área administrativa podamos modificar o agregar contenido para el apartado acogida de la web.

En este sprint como objetivo se traza el permitir agregar más contenido y modificarlo, pero esta vez para el área de extraescolares, comisiones y acogida. Viendo que el desarrollo del sprint fue satisfactorio, se puede dar por cumplido el objetivo.

SPRINT 10:

Historia 28: como parte del área administrativa nos sería útil tener un apartado que nos permita modificar o agregar contenido al apartado de la web tardes de junio.

Historia 29: dentro del área administrativa sería útil tener un lugar desde donde modificar o agregar contenido nuevo a la página de casal de verano.

Historia 30: sería de utilidad para nosotras tener un apartado dentro del área administrativa que permita agregar contenido o editar el contenido de la página equipamiento.

En este sprint el objetivo asumido es el de poder agregar, modificar contenido en los apartados tardes de junio, casal de verano y equipamiento. Dicho lo anterior se puede dar por cumplido parte de los objetivos. No en su totalidad por la falta de información que existe aun para el apartado del equipamiento.

SPRINT 11:

Historia 31: poder modificar el contenido o agregar nuevo contenido a la página actas desde el área administrativa.

Historia 32: tener apartado dentro del área administrativa para poder modificar o agregar nuevo contenido a la página de presupuestos.

Historia 33: tener un formulario de inscripción a las comisiones para nuevos afiliados a las comisiones

En este sprint el objetivo definido por las tres historias es poder afiliar tutores legales de la escuela Estalella a las distintas comisiones desplegadas por la AFA durante el año, estas comisiones brindar soporte activo a la AFA en el momento del despliegue de actividades o del soporte necesario para cuando se hacen ciertas búsquedas de proveedores o negociaciones con distintos proveedores que permiten agregar valor a las actividades desarrolladas por la AFA para el alumnado. Así también se busca el poder modificar o agregar contenido a las partes como actas y presupuestos que contiene el sitio web. Revisando el sprint y el desarrollo activo, se puede dar como cumplidos estos objetivos.

SPRINT 12:

Historia 34: nos sería de utilidad para nosotros poder tener un apartado para poder editar y crear contenido para la página de estatutos.

Historia 35: cuando entre al apartado administrativo quisiera poder agregar el contenido de la página Que somos.

Historia 36: tener un formulario de inscripción a las extraescolares para los alumnos.

El objetivo buscado en este sprint es el de poder modificar, agregar nuevo contenido, así como brindar la posibilidad de la inscripción a las extraescolares mediante el sitio web. Revisado este sprint y dado como hecho según su definición. Se puede dar por cumplido los objetivos descritos.

SPRINT 13:

Historia 37: quisiera poder desde el área administrativa poder modificar el contenido de la página donde las familias pueden hacerse socios de la AFA. Modificar el prólogo que describe porque es bueno pertenecer a la AFA.

Historia 38: tener un formulario de inscripción a la AFA para nuevos afiliados.

Historia 39: nos agradecería que la página se pueda ver en todo tipo de dispositivos porque no siempre se tiene un ordenador a la mano, pero si un móvil o una Tablet.

El objetivo definido por las historias tratadas en el sprint es, la inscripción de nuevos afiliados a la AFA, la modificación y agregación de nuevo contenido dentro del apartado hacerse socio de la AFA y por otro lado el que la aplicación web sea responsive. Que pueda verse en cualquier dispositivo de una forma práctica y manejable.

SPRINT 14:

Historia 40: quisiera poder tener dentro del área administrativa un lugar donde poder ver los mails que nos han enviado y así poder tener todo en un solo lugar donde pueda consultar esta información.

Historia 41: nos gustaría poder tener un apartado dentro del área administrativa desde donde pueda consultar los inscritos en las distintas comisiones.

Historia 42: sería de utilidad poder tener un lugar dentro del área administrativa desde donde poder consultar los afiliados a la AFA.

El objetivo de principal poder realizar consultas que permitan tomar decisiones sobre las comisiones, afiliados y mails de consultas enviados a la AFA. Este objetivo no se ha podido cumplir por falta de tiempo. Sin embargo, se pondrá como próximos objetivos o nuevas funcionalidades a desarrollar dentro de la aplicación.

SPRINT 15:

Historia 43: nos ayudaría tener en el área administrativa un apartado que permita ver la información de los alumnos que se anotan en las extraescolares.

Historia 44: cuando se inscriban en una extraescolar se ha de tener siempre las plazas que queden en todo momento libres.

Historia 45: dentro del área administrativa nos sería útil el poder tener un lugar desde donde poder hacer un mailing a todas las familias afiliadas al AFA.

El objetivo buscado en este sprint se encuentra en poder realizar comunicaciones a todos los afiliados usando un mailing que permita enviar dicha información. Consultas sobre las extraescolares. Estos objetivos no se han cumplido por tiempo, se harán dentro de las nuevas funcionalidades a desarrollar en el crecimiento del sitio web.

SPRINT 16:

Historia 46: Nos agradecería poder adjuntar imágenes de los eventos realizados en las noticias que presentamos en el apartado noticias al darle un clic a la noticia.

Historia 47: poder modificar o agregar más imágenes al carrusel de la web de la AFA.

Historia 48: poder tener un enlace al Instagram de la escuela.

En este sprint existe como objetivos el nexo entre la aplicación web y los canales alternos como pueden ser Instagram, Facebook, etc. Así también poder modificar las imágenes representativas de la AFA en el menú principal, también poder modificar y agregar imágenes a cada evento y noticia que se despliega en estos apartados. Los objetivos no han sido cumplidos en su totalidad por falta de canales alternativos y por el por falta de tiempo. Sin embargo, se usarán como nuevas funcionalidades para el mantenimiento de la aplicación.

SPRINT 17:

Historia 49: poder agregar más usuarios con determinados perfiles que puedan tener funciones dentro de la web.

Historia 50: poder realizar consultas de los datos de las extraescolares como por ejemplo ¿Qué extraescolar es la más demandada?

Historia 51: almacenar datos de los usuarios que no han podido obtener una plaza en alguna extraescolar para poder tener encuentra la creación de nueva extraescolar bajo la misma temática.

El objetivo de este sprint el poder implementar funcionalidades que permitan tomar decisiones sobre que extraescolares abrir y cuantas respecto a futuros años curriculares, así como también permitir tener distintos perfiles de usuarios administrativos para que manejen determinadas características dentro del sitio web.

Estos objetivos no han sido cumplidos, sin embargo, se retomarán dentro de las nuevas funcionalidades que se realizarán en el mantenimiento de la aplicación.

SPRINT 18:

Historia 52: poder exportar los resultados de las consultas a formato xls.

Historia 53: tener un formulario para que los tutores puedan enviarnos como datos adjuntos los pagos al banco para la inscripción de la extraescolar y así poder tener almacenado este comprobante dentro de la inscripción.

Historia 54: poder tener la posibilidad de agregar y/o modificar el menú principal.

El objetivo agregar más autonomía a la AFA para la creación de nuevos apartados, retomar un cambio dentro de las inscripciones de las extraescolares y así avanzar a que se automatice cada vez más el proceso, así también el poder exportar las consultas en formato xls para poder permitir al usuario sentirse cómodo al manejar los datos en una herramienta de uso cotidiano. Objetivo no cumplido, pero se tomará en cuenta dentro de las nuevas funcionalidades en el mantenimiento de la aplicación.

Después de examinar cada sprint y haber analizado en cada caso si se han cumplido o no los objetivos necesarios para la aplicación web se puede decir que los objetivos en su gran mayoría han sido cumplidos. Sin embargo, existen algunos que no se han podido cumplir por tiempo, pero se ha indicado que se realizará un mantenimiento a la aplicación para poder cumplir y finalizar los objetivos no cumplidos. De esta manera el producto será plenamente utilizado por la AFA. Este TFG ha permitido conocer con mayor amplitud lo que es la AFA, cuáles son sus labores y permitir trabajar con nuevas tecnologías. El aprendizaje ha sido completo, pues no solo después del desarrollo se tiene un conocimiento de las nuevas tecnologías usadas, sino que también se permite conocer el valor de las metodologías usadas durante el desarrollo, la experiencia ganada durante este desarrollo ha sido satisfactoria y me permitirá afrontar nuevos retos en mi vida profesional.

6. Bibliografía

Documentación oficial de React: interactividad a una página HTML

<https://es.reactjs.org/docs/getting-started.html>

Documentación oficial de Node: módulos integrados

<https://nodejs.org/en/docs/>

Nielsen Norman Group: Fuente de información de usabilidad para el diseño de la interfaz de usuario

<https://www.nngroup.com/articles/ten-usability-heuristics/>

Documentación oficial de Firebase: desarrollo de aplicación web

<https://firebase.google.com/docs>

Documentación oficial de Aiox: Liberia Axios

<https://firebase.google.com/docs>

Documentación oficial de React: enrutamiento declarativo para aplicaciones React a cualquier escala

<https://reactrouter.com/docs/en/v6>

Documentación oficial de Bootstrap: herramientas frontend

<https://getbootstrap.com/docs/5.2/getting-started/introduction/>

Documentación oficial de Awesome: Documentación de API

<https://awesomewm.org/doc/api/>

Firebase Authentication SDK

<https://firebase.google.com/docs/auth>

Documentación oficial de Visual Studio Code

<https://code.visualstudio.com/docs>

Información de Wikipedia: NoSQL

<https://es.wikipedia.org/wiki/NoSQL>

Información de Wikipedia: Interfaz de programación de aplicaciones

https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones

Documentación oficial de CSS: Renderizado

<https://developer.mozilla.org/es/docs/Web/CSS>

Documentación oficial de CSS: propiedades CSS

<https://www.w3schools.com/cssref/>

Patrones de diseño: qué son y por qué debes usarlos

<https://www.genbeta.com/desarrollo/patrones-de-diseno-que-son-y-por-que-debes-usarlos>

Revista electrónica La comunicación familia-escuela: el uso de las TIC en los centros de primaria

[La comunicación familia-escuela: el uso de las TIC en los centros de primaria \(redalyc.org\)](http://redalyc.org)

Observatorio tecnológico: Diseño de páginas web educativas

[Diseño de páginas web educativas en centros educativos | Observatorio Tecnológico \(educacion.es\)](http://educacion.es)