



AWESOMETRACKER

Versión 0.6

Descripción

Proyecto de fin de ciclo DAW IES San Clemente 2018-2020

Pablo Liste Cancela
ppabli12@gmail.com | a18pablo1c@iessanclemente.net

Índice

Estudio Preliminar	4
Introducción	4
Convenciones	4
Audiencia	5
Objetivo del producto	5
Descripción general	5
Perspectiva	5
Requisitos del negocio	6
Usuarios	6
Logs	6
Aplicaciones	6
Apps	7
Inicios de sesión	7
Peticiones a la API	7
Peticiones a la web	7
Posts	8
Categorías	8
Tipos de usuario	9
Tipos de apps	10
Entorno operacional de la aplicación	11
Planificación y costes	12
Estimación inicial	12
Calendario inicial	12
Estimación coste inicial	12
Tabla de costes	12
Costes desarrollo	12
	1

Costes de mantenimiento	13
Coste por actividad	13
Análisis	14
Funciones	14
Función 1: Dar de alta al usuario	14
Función 2: Dar de baja al usuario	15
Función 3: Modificar un usuario	15
Función 4: Mejorar la categoría de un usuario	15
Función 5: Iniciar sesión	16
Función 6: Dar de alta un log	16
Función 7: Eliminar un log	16
Función 8: Dar de alta una aplicación	16
Función 9: Eliminar una aplicación	17
Función 10: Modificar una aplicación	17
Función 11: Dar de alta una App	17
Función 12: Dar de baja una App	17
Función 13: Modificar una App	18
Función 14: Mejorar la categoría de una App	18
Recursos necesarios	19
Recursos materiales	19
Hardware	19
Software	20
Servicios contratados	20
Recursos humanos	21
Diseño	22
Base de datos	22
Modelo entidad relación	23

Modelo relacional	24
Análisis	25
Mockups	29
Diseño general	29
Index	32
Inicio de sesión (Login)	34
Recuperación de cuenta 1 (forgot)	36
Recuperación de cuenta 2 (recover)	37
Creación de cuenta	38
Perfil o App	39
Data 1	40
Data 2	42
Codificación y pruebas	44
Codificación	44
Pruebas	45
Futuras mejoras	45
Diseño	45
Funcionalidades	46
Aplicación de escritorio	46
Bibliografía	47
Anexos	48
Anexo 1	48
Anexo 2	48
Anexo 3	48
Anexo 4	48

Estudio Preliminar

Introducción

Este documento presenta un servicio que permite a los usuarios ver datos relacionados con el uso que hace del ordenador similar a la aplicación de tiempo de uso disponible en los productos de Apple. Esta idea nace de la falta de algo similar en el ecosistema de Windows/Linux y que impide a los usuarios saber cuánto tiempo pasan trabajando en el ordenador o que aplicaciones no han usado últimamente entre muchas otra, así como la creciente demanda en todo el mundo de seguridad y salud digital.

Como concepto el usuario a través de la aplicación web se descargará una aplicación que instalaría en su equipo la cual registraría cuanto tiempo pasa en cada aplicación y mandaría esos datos al servidor. Luego a través de la misma aplicación web que uso para descargar el software el usuario podría ver los datos recopilados por la aplicación de escritorio en forma de tablas, gráficos y demás representación además realizar una serie de acciones como crear y modificar categorías, hacer uso de los datos recopilados mediante una API, obtener recomendación y otras opciones.

Convenciones

Log – Cada uno de los registros de tiempo de uso de una aplicación que hace la aplicación y que están almacenados en la base de datos de la aplicación.

App – Cada uno de los tokens proporcionados que permiten el uso de la API.

Diff – Tiempo mínimo para que se registre un log en el sistema. Individual y personalizable por cada usuario.

Post – Cada una de las noticias que se muestran en la página. Estas incluirán información sobre cambios, novedades o advertencias.

Estatus – Estado de un recurso. Se emplea para los usuarios o apps. Permite o no hacer uso de los recursos.

Visibilidad – Privacidad de la cuenta de usuario o app. Permite o no a los demás usuarios visualizar los datos tanto propios como relacionas de dicho recurso tanto.

Audiencia

El público objetivo de nuestra aplicación comprende a usuarios que tienen interés en llevar el control del tiempo que emplean usando el ordenador con el fin de no solo saber que aplicaciones usan y el tiempo que emplean si no que busquen mejorar sus rendimientos y salud digital en base a los datos y consejos de nuestra aplicación.

También puede resultar atractiva a empresas que busquen un sistema de bajo coste y código libre para monitorizar el uso de los equipos informáticos, que aplicaciones se usan en estos, así como conocer el uso que hacen los empleados de los mismos.

A su vez al tener disponible los datos recopilados accesibles desde una API también puede interesar a empresas que busquen obtener información para sus negocios o proyectos.

Objetivo del producto

Ayudar a los usuarios que busquen mejorar su salud digital permitiéndoles conocer el tiempo de uso y aplicaciones que emplean en su día a día mediante intuitivos gráficos, tablas y consejos adaptados. También ayuda a empresas que busquen monitorizar el uso de los equipos y aplicaciones en el centro de trabajo mediante una aplicación de escritorio optimizada y Open Source o bien obtener información sobre usuarios de los equipos informáticos recurriendo a la API que implementara la aplicación.

Descripción general

Perspectiva

Permite al usuario la consulta de información actualizada sobre su tiempo de uso con simplemente iniciar sesión en nuestra página web. A su vez también permite a las empresas un rápido despliegue y gestión de cuentas gracias la implementación de la API y Apps al igual que el escaso consumo de espacio y cómputo de nuestra aplicación la hace ideal para implementar el equipo de escasos recursos.

Requisitos del negocio

En un inicio la aplicación registra: usuarios, logs, aplicaciones, apps, inicios de sesión, peticiones a la API, peticiones a la aplicación web y posts.

Usuarios

Dato	Tipo
Código de usuario	Numérico
Código de la aplicación que creó el usuario	Numérico
Código de estatus	Numérico
Visibilidad	Texto
Diff	Numérico
Nick	Texto
Contraseña	Texto
Email	Texto
Código de categoría	Numérico
Nombre	Texto
Apellidos	Texto
Fecha de registro	Fecha y hora
Fecha de actualización	Fecha y hora
Fecha de cumpleaños	Fecha y hora
URL de la imagen	Texto
Códigos de recuperación	Texto

Logs

Dato	Tipo
Código del log	Numérico
Código de usuario que creó el log	Numérico
Código de la aplicación	Numérico
Fecha de inicio	Fecha y hora
Fecha de fin	Fecha y hora
Duración	Numérico

Aplicaciones

Dato	Tipo
Código de la aplicación	Numérico
Código del usuario al que pertenece	Numérico
Categoría	Texto
Aplicación	Texto
Fecha de registro	Fecha y hora
Fecha de actualización	Fecha y hora

Apps

Dato	Tipo
Código de la app	Numérico
Código del usuario al que pertenece	Numérico
Código de status	Numérico
Visibilidad	Texto
Token	Texto
Código de categoría	Numérico
Nombre	Texto
Descripción	Texto
URL de la imagen	Texto
Fecha de creación	Fecha y hora
Fecha de modificación	Fecha y hora

Inicios de sesión

Dato	Tipo
Código de sesión	Numérico
Código del usuario que inicio sesión	Numérico
Fecha de inicio	Fecha y hora
Tipo de sesión	Texto
Dispositivo	Texto

Peticiones a la API

Dato	Tipo
Código de petición	Numérico
Código de la app	Numérico
IP del equipo	Texto
URL solicitada	Texto
Método	Texto
Fecha	Fecha y hora

Peticiones a la web

Dato	Tipo
Código de la petición	Numérico
IP del equipo	Texto
URL solicitada	Texto
Método	Texto
Fecha	Fecha y hora

Posts

Dato	Tipo
Código del post	Numérico
Código del usuario que lo creó	Numérico
Título	Texto
Contenido	Texto
Fecha de registro	Fecha y hora
Categoría del post	Numérico

La aplicación permitirá a través de una aplicación de escritorio registrar el uso que hace el usuario de cada aplicación y luego a través de una aplicación web este tendrá acceso a ver esos datos obtenidos de una manera más rápida, sencilla y clara a su vez se le permitirá realizar modificaciones sobre estos como modificar el perfil, crear o modificar sus aplicaciones o apps, mejorar de categorías sus usuarios o apps permitiéndoles realizar más acciones así como obtener acceso a la API o a los post de nuestra aplicación.

Los usuarios que usen nuestra aplicación proporcionarán información referente al dispositivo desde donde se conectan, fecha y hora del sistema, así como información personal como nombre, apellidos o correo electrónico. Todas las políticas y términos de uso están reflejados en la propia página.

Categorías

En el sistema tanto usuario como apps tienen una categoría asignada. Esta categoría está indicada mediante un código el campo de **categoryCode** de cada recurso. Dependiendo de la categoría a la que pertenezca el recurso eliminará en mayor o menor medida las restricciones de uso, así como permitirles realizar ciertas acciones extra. En caso de que se quiera realizar un cambio de categoría se tendrá que realizar un único pago de la cantidad marcada en el precio de la categoría. A continuación, se detallan los datos de cada categoría.

Nombre	Precio (€)	Código
Estándar (Por defecto)	0	7
Premium	20	6
Super	35	5
Ultra	50	4
Awesome	100	3
Almost admin	200	2
Admin	0	1

Tipos de usuario

Nuestra aplicación distingue diferentes tipos de usuarios según su categoría. Todos los usuarios pertenecen a una categoría la cual está indicada en el propio recurso y que indica una relación con la tabla de categorías de usuario. Cada categoría de usuario permite a estos realizar ciertas acciones en mayor o menor medida o realizar acciones más avanzadas como las de administración, actualmente las categorías de usuario existentes son estas:

Nombre	Apps máximas
Estándar (Por defecto)	2
Premium	5
Super	10
Ultra	15
Awesome	20
Almost admin	Ilimitadas
Admin *	Ilimitadas

* Es la máxima categoría de usuario permitida y hace que quien la posea tenga acceso a todas las opciones de administración disponibles en la aplicación web, así como la eliminación del límite a apps de su cuenta. Esta categoría sólo puede ser asignada por otros usuarios administradores.

Todos los usuarios tienen acceso a la creación de apps, aplicaciones, así como al uso de la aplicación de escritorio y administración de sus datos.

Tipos de apps

Ya que nuestra aplicación no solo tiene usuarios que acceden al servicio a través de la aplicación web, sino que también tiene una API a través de la cual las diferentes apps pueden realizar acciones también se ha creado un sistema de categorías para estas que permitirá a la aplicación gestionar mejor el uso de los recursos, así como facilitar la obtención de ingresos al poder monetizar el uso de sus recursos. Cualquier categoría mencionada permitirá a la app que la posea realizar las mismas acciones sin embargo la cantidad de estas se verá limitada por el número máximo de petición a la API marcada en su categoría.

Nombre	Peticiones diarias máximas
Estándar (Por defecto)	5
Premium	20
Super	50
Ultra	100
Awesome	200
Almost admin	Ilimitadas
Admin *	Ilimitadas

* Es la máxima categoría para una app. Esta elimina el máximo número de peticiones a la API diarias, así como de cualquier comprobación de permisos y seguridad. Esta categoría únicamente es empleada por la app de sistema. A su vez sólo puede ser asignada por otra app que la posea o por un usuario con categoría admin.

Entorno operacional de la aplicación

Para acceder a la aplicación web desde donde se tendrá acceso a la información recopilada por la aplicación de escritorio únicamente se necesitará un dispositivo con navegador web y acceso a internet. Los navegadores empleados para el desarrollo y donde se ha comprobado el correcto funcionamiento de la aplicación son los siguientes: Google Chrome 21+, Opera 12.1+, Safari 6.1+, Mozilla Firefox 28+, Microsoft Edge 12+, IE10+ y Chromium 22+. La velocidad de internet recomendada es de 2 Mbps.

A su vez en lo que abarca a la aplicación de escritorio el equipo anfitrión únicamente tendrá que disponer de un sistema operativo Windows o macOS tanto en versión de x86 como x64. Este equipo deberá contar con un mínimo de 20 MB para la aplicación de escritorio, así como 10 MB de RAM disponible.

En caso de querer hacer uso del código de la aplicación para desarrolló los módulos necesarios, así como los requisitos de hardware son los siguientes:

Requisitos generales independientes del sistema operativo empleado:

- 2 Mbps de velocidad de conexión
- 10 MB de RAM disponible
- 36,5 KB de almacenamiento disponible
- Python 3.6
- Módulo de Python Requests 2.20

Para Windows se necesitará a mayores de lo anterior el módulo de Python PyWin32 225

Para macOS se necesitará a mayores de lo anterior el módulo de Python PyObjC 6.1

Planificación y costes

Estimación inicial

Calendario inicial

Actividad	Inicio	Duración (Días)	Fin
Estudio preliminar	16/03/2020	10	26/03/2020
Análisis de requisitos	26/03/2020	15	10/04/2020
Diseño	10/04/2020	20	30/04/2020
Codificación y pruebas	30/04/2020	45	14/06/2020

Estimación coste inicial

Se estima que los gastos para el desarrollo de esta aplicación no superen los 1300 euros mensuales. Se prevé un desarrollo de unos 3 meses de duración por lo que los gastos totales no deberían superar los 4000 euros. Estos gastos abarcan la contratación del servidor donde se ejecuta la aplicación y donde se guardan los datos recopilados siendo este de un total de 50 euros al mes, así como el nombre de dominio que emplea con un coste de 15 euros anuales, conexión a internet privada que nos permita agilizar el desarrollo 70 euros mensuales, así como 90 euros que corresponden a la parte proporcional del coste del equipo personal donde se va a desarrollar. En cuanto a los costes de suministros solo vamos a tener en cuenta los suministros eléctricos los cuales se estiman que serán de unos 50 euros mensuales, no se tienen en cuenta gastos de alquiler, agua o alimentación. En cuanto al sueldo del programador este será de 1000 euros.

Para minimizar los gastos se hará uso de aplicaciones Open Source o gratuitas como Libre Office, GIMP o Visual Studio Code.

Tabla de costes

Costes desarrollo

Concepto	Coste mensual (€)	Coste total proyecto (3 meses) (€)
Servidor	50	150
Conexión a internet	70	210
Gastos de dominio	1,25	3,75
Coste equipo	90	270
Coste suministros	50	150
Salario programador	1000	3000
Aplicaciones	0	0
Total	1275	3795

Costes de mantenimiento

Concepto	Coste mensual (€)	Coste anual (€)
Servidor	50	600
Conexión a internet	70	840
Gastos de dominio	1,25	15
Coste suministros	50	600
Total	1275	2055

Coste por actividad

Actividad	Duración (horas)	Coste (€)		Coste total (€)
		Personas	Recursos materiales	
Estudio preliminar	80	333	77	400
Análisis de requisitos	120	500	130	636
Diseño	160	667	178	849
Codificación y pruebas	360	1500	410	1910
Total, proyecto (€)		3000	795	3795

Análisis

Funciones

A continuación, se listan todas las acciones que se permiten realizar tanto a las apps como al usuario del sistema a través de nuestra aplicación. La mayoría están disponibles para su uso a través de la API o de la versión web.

Función 1: Dar de alta al usuario

Requerirá de los siguientes datos para completarse:

- Código de usuario por defecto será incremental y autogenerado por la propia aplicación.
- Código de la aplicación que crea el usuario.
- Nick.
- Contraseña la cual será procesada antes de guardar en la base de datos y deberá contar con al menos 8 caracteres incluyendo letras (mayúsculas y minúsculas), números y caracteres no numéricos.
- Email.
- Nombre de usuario.
- Apellidos del usuario.
- Fecha de registro será la hora actual del sistema.
- Fecha de última actualización la cual será la hora actual del sistema.
- Fecha de cumpleaños.
- Diff por defecto tiene el valor de 1.
- URL de la imagen del usuario en caso de no proporcionar ninguna por defecto se utilizará la básica de la aplicación¹.
- Código de estatus del usuario por defecto 1.
- Código de categoría del usuario por defecto se asigna la mínima categoría, es decir el código 7.
- Visibilidad del usuario pudiendo ser public o private se asigna por defecto public.

Algunos de campos tienen un valor por defecto ya que dependiendo de quién cree el usuario estos se podrán añadir o no.

¹ <https://awesometracker.ddns.net/userImg/default.png>

Función 2: Dar de baja al usuario

Requerirá de los siguientes datos para completarse:

- Código del usuario a eliminar.

Función 3: Modificar un usuario

Para poder modificar un usuario únicamente hará falta proporcionar el código del usuario que queremos modificar, así como las propiedades de este que queremos cambiar. Las propiedades modificables del usuario son todas las que posee menos propio código, fecha de registro del usuario, así como la fecha de última modificación.

Ciertos cambios no están permitidos dependiendo de quién modifique el usuario por ejemplo si quien modifica el usuario es el mismo este no podrá cambiar el código de la app que lo creó, código de categoría, código de usuario. Mientras que en si quien está modificando el usuario es la app que lo creo esta puede modificar todos los campos menos el código de categoría y el código de usuario. Ya como único caso en caso de que una app o usuario administrador sea quien modifique un usuario este podrá modificar todos los valores salvo el código de propio usuario.

Función 4: Mejorar la categoría de un usuario

Ya que nuestra aplicación hace uso de un sistema de categorías de usuario el sistema también permite al usuario realizar el proceso de mejorar su usuario mediante un solo pago para mejorar las capacidades de este. Para ello el sistema hace uso de la API de PayPal y por defecto el único parámetro que se tienen que proporcionar es el código de la nueva categoría. Por defecto esta acción únicamente está presente en la aplicación web.

Función 5: Iniciar sesión

Requerirá de los siguientes datos para completarse:

- Email o nombre usuario,
- Contraseña.

El hecho de iniciar sesión implica un registro en la base de datos con los siguientes datos:

- Código de sesión por defecto un valor incremental autogenerado por el sistema.
- Código del usuario que ha iniciado sesión.
- Fecha de inicio.
- Tipo de sesión (cierre o inicio).
- Dispositivo desde donde se realizó la acción.

Función 6: Dar de alta un log

Requerirá de los siguientes datos para completarse:

- Código del log el cual inicialmente será un valor incremental y autogenerado por el sistema.
- Código del usuario que crea el registro.
- Fecha de inicio.
- Fecha de fin.

Función 7: Eliminar un log

Requerirá de los siguientes datos para completarse:

- Código del log el cual queramos eliminar.

Función 8: Dar de alta una aplicación

Requerirá de los siguientes datos para completarse:

- Código de la aplicación por defecto es un valor incremental y autogenerado por el sistema.
- Código del usuario que da de alta la nueva aplicación.
- Nombre de la categoría.
- Nombre de la aplicación.

Función 9: Eliminar una aplicación

Requerirá de los siguientes datos para completarse:

- Código de la aplicación que queramos eliminar.

Función 10: Modificar una aplicación

Requerirá de los siguientes datos para completarse:

- Código de la aplicación que queramos modificar.
- Código del usuario propietario.
- Categoría.
- Nombre.

Función 11: Dar de alta una App

Requerirá de los siguientes datos para completarse:

- Código de la app por defecto este será un valor incremental y autogenerado por el sistema.
- Código de usuario al que pertenece.
- Nombre de la app.
- Descripción de la app.
- Código de estatus.
- Visibilidad.
- Token de acceso a la API este será un valor autogenerado.
- Código de categoría a la que pertenece.
- Fecha de registro la cual será la hora del sistema en el momento de creación.
- Fecha de última modificación la cual será la hora del sistema en el momento de creación
- URL de la imagen de la app por defecto hará uso de la imagen por defecto².

Función 12: Dar de baja una App

Requerirá de los siguientes datos para completarse:

- Código de la app que se quiere eliminar.

² <https://awesometracker.ddns.net/userImg/default.png>

Función 13: Modificar una App

Para poder modificar una app únicamente se tendrá que adjuntar el código de la app que queremos modificar. Todas las propiedades de esta son modificables exceptuando el propio código, fecha de registro del usuario, así como la fecha de última modificación.

El token de la app no es modificable desde la propia API se tendrá que acceder a la aplicación web con el usuario propietario y realizar el cambio. A su vez únicamente se podrá cambiar el código de usuario al que pertenece la app cuando quien la modifique sea un usuario administrador a través de la aplicación web.

Función 14: Mejorar la categoría de una App

Al igual que con los usuarios nuestra aplicación desarrolla un sistema de categorías para las apps. Para ello el sistema hace uso de la API de PayPal y por defecto el único parámetro que se tienen que proporcionar es el código de la nueva categoría. Por defecto esta acción únicamente está presente en la aplicación web.

Aparte de todas las funciones anteriormente mencionadas el sistema permite funciones de listado que están disponibles a través de la API y que son las empleadas para obtener ciertos datos como los mostrados en el panel de control de la aplicación web, etc.

Ciertos valores que son necesarios para realizar alguna de las acciones anteriores son autogenerados o bien los recoge el sistema automáticamente desde la base de datos, dispositivo del usuario o sesión y no son necesarios.

El uso de la API también genera un registro de peticiones que nos permite llevar un control de las llamadas. Estos registros cuentan con los siguientes campos: código de llamada, código de la app que ha hecho la petición, URL solicitada, método, y fecha y hora de la petición, así como la IP de la máquina desde la cual se realizó la acción.

Cabe destacar que en las funciones de adición, modificación o borrado se comprueba que los valores proporcionados son correctos es decir no hay conflicto con otros ya existentes, así como se comprueba que quien solicita esa acción tiene los permisos necesarios para realizarla.

Recursos necesarios

Es esta sección detallaremos los equipos informáticos, servicios y aplicaciones que se usarán para el desarrollo de la aplicación, así como los costes, en caso de que existan, de cada uno de ellos.

Recursos materiales

Hardware

Empezaremos por los más importante que es el equipo donde el programa va a desarrollar la aplicación. En un inicio y dada la corta envergadura del proyecto únicamente se hará uso de 2 equipos informáticos. El primero mi ordenador personal de sobremesa, así como sus 2 pantallas ya que considero que proporciona la potencia, versatilidad y rendimiento necesario para el desarrollo del proyecto además como dispositivo de apoyo se hará uso de un MacBook Pro de 2017 en el cual podremos probar la aplicación de escritorio par dispositivos MacOS.

Continuamos por otro de los dispositivos más importantes hoy en día³, el o los dispositivos móviles. Estos nos permitan comprobar el rendimiento, usabilidad, accesibilidad y comportamiento de la aplicación web en estos dispositivos de pantallas y potencia más reducida. Para realizar estas pruebas y al igual que en el caso anterior vamos a hacer uso de los dispositivos personales que tengo a mi disposición que son varios terminales Android como son mi OnePlus 6 y 3 así como otro Samsung Galaxy S8 a su vez y para testear el sitio desde un dispositivo de Apple con su navegador propietario haremos uso de un iPhone 6. Para poder testear la aplicación en dispositivos de pantalla intermedias entre un móvil y un portátil haremos uso de una Tablet Android Samsung Galaxy Tab S2, así como un iPad de 2018 estos también dispositivos personales.

³ <https://www.broadbandsearch.net/blog/mobile-desktop-internet-usage-statistics>

Software

En cuanto a las aplicaciones que usaremos para desarrollar y testear nuestra aplicación tenemos que intentaremos primar las aplicaciones de código abierto y gratuitas como son en la categoría de IDEs: Visual Studio Code, Atom o Notepad En nuestro caso haremos uso de Visual Studio Code. En la categoría de navegadores y al tratarse de aplicaciones web tenemos que hacer uso del mayor número de estos para asegurarnos que nuestro sitio funciona correctamente en cuanto a usabilidad, accesibilidad y funcionalidad de cara a todos los usuarios independiente de navegador que usen. Para esto haremos uso de Google Chrome, Opera, IE, Microsoft Edge, Mozilla Firefox, Safari, Chrome entre otros. Nuestra aplicación hace un uso relativamente extenso de las bases de datos relaciones MariaDB para ello tenemos que tener disponible algún software que nos permita trabajar en este sistema de manera externa al terminal del propio servidor para ello haremos uso de MySQL Workbench. Para el testeo de la API haremos uso de la aplicación de escritorio Postman. A su vez y para el desarrollo de la parte del cliente instalaremos en nuestras máquinas de desarrollo vue-cli que nos permite un desarrollo más rápido y eficiente. Para ello instalaremos NodeJS en nuestras máquinas, así como los diferentes módulos como son Bootstrap, vue-cli, así como el procesador de ficheros sass.

Servicios contratados

En relación a todo lo anterior tenemos el servidor donde se estará ejecutando nuestra aplicación, bases de datos, contendrá nuestro fichero documentos, etc. Para este equipo haremos uso del cloud computing tan de moda últimamente⁴ ya que nos permite un despliegue extremadamente rápido, eficiente y económico de nuestra aplicación en estas etapas iniciales. Como proveedor de este servicio haremos uso de Google Cloud Computing una de las mejores soluciones actualmente en internet gracias a sus precios tan competitivos. El equipo contratado es lo que denominan n1-standard-2 que cuenta con 2 virtual CPUs y 7,5 GB de memoria RAM siendo ésta la que más necesitara nuestra aplicación. Esta máquina contará con las configuraciones requeridas por la aplicación como la apertura de ciertos puertos como el 3306, 80 o 443, así como la instalación de determinados programas o complementos como NodeJS para el desarrollo de la aplicación o MariaDB para el desarrollo de las bases de datos

⁴ <https://channels.theinnovationenterprise.com/articles/why-cloud-computing-is-so-popular-and-how-it-transforms-business>

relacionales de nuestra aplicación, así como módulos de estos. En nuestro caso y gracias a hacer uso de NodeJS y de su gestor de paquetes nuestra aplicación tanto del lado del cliente como del lado del servidor apenas van a necesitar de recursos que nos estén localizados en nuestro servidor es por eso que vamos a prescindir de CDN y en su lugar, siempre que sea posible, vamos a realizar la instalación de estos mediante módulos de NodeJS y él envió de los recursos junto con la petición como por ejemplo ficheros de CSS o JS. En cuanto a los módulos empleados haremos uso de body-parser, express-session, cookie-parser, dotenv, http, https, útil, request-promise, request, express, nodemailer, formidable, fs, crypto y paypal. A su vez para el uso del sistema de login con Google haremos uso de su CDN.

Dentro de los recursos y configuraciones más importantes para el desarrollo de nuestra aplicación solo nos queda la contratación de nombre de dominio por el cual los usuarios y sistemas podrán comunicarse con nuestro servidor sin ser por la dirección IP. Este tipo de servicios es muy recomendable ya que de cara al programador es más sencillo apuntar a un nombre de dominio que no a una IP ya que esta última puede variar pudiendo dejar sin servicio a los usuarios y lo más importante de cara al usuario final de nuestra aplicación ya que resulta mucho más fácil recordar un nombre que no un conjunto de números. Para este servicio estableceremos una relación comercial con una empresa propia de Galicia como es Dinahosting que nos proporcionan servicios de hosting con terminación .com como es la que deseamos por un módico precio de unos 15 euros anuales.

Recursos humanos

Una vez tenemos todos los sistemas y servicios que va a requerir tanto el desarrollo de nuestra aplicación como la propia aplicación solo nos queda el personal humano encargado de desarrollarla, actualizarla y mantenerla operativa los programadores y miembros del equipo de IT. Como ya hemos visto antes este es un proyecto pequeño por lo que el equipo de IT únicamente contará con un único miembro que se encargará de las tareas de codificación, despliegue y mantenimiento de los sistemas informáticos.

Diseño

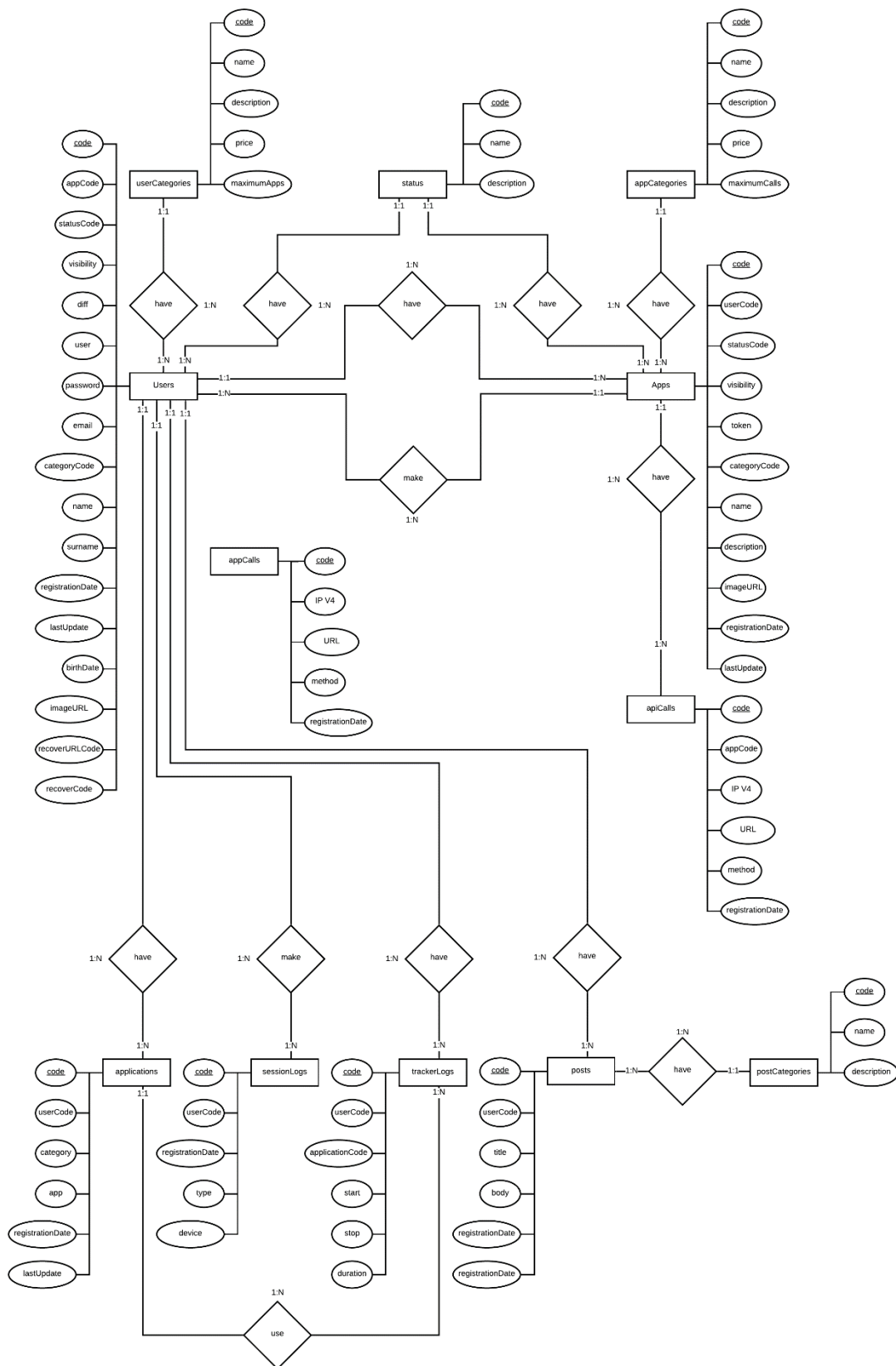
Base de datos

Toda aplicación necesita datos para poder funcionar y la nuestra no va a ser una excepción ya que recopiló datos de uso de las aplicaciones, sesiones de los usuarios, categorías, etc. Todos estos datos necesitan ser almacenados y procesados para poder ser mostrados al usuario de la mejor manera posible para ello haremos uso de las bases de datos en concreto haremos uso del sistema de gestión de datos MariaDB. Hemos elegido este sistema y no otro ya que nuestra aplicación contará con una gran cantidad de tablas y relaciones entre ellas así que necesitábamos hacer uso de un sistema de bases de dato relacionales que nos permita obtener la información completa a pesar de estar en diferentes tablas. Dentro de todos los sistemas de gestión de bases de datos relacionales que existen hemos optado por MariaDB dada su mejor soporte a motores, así como su mejor rendimiento⁵ y amplias funcionalidades aparte de que como ya hemos visto en apartados anteriores intentaremos priorizar el uso de software libre y este MariaDB se distribuye bajo una licencia GPL.

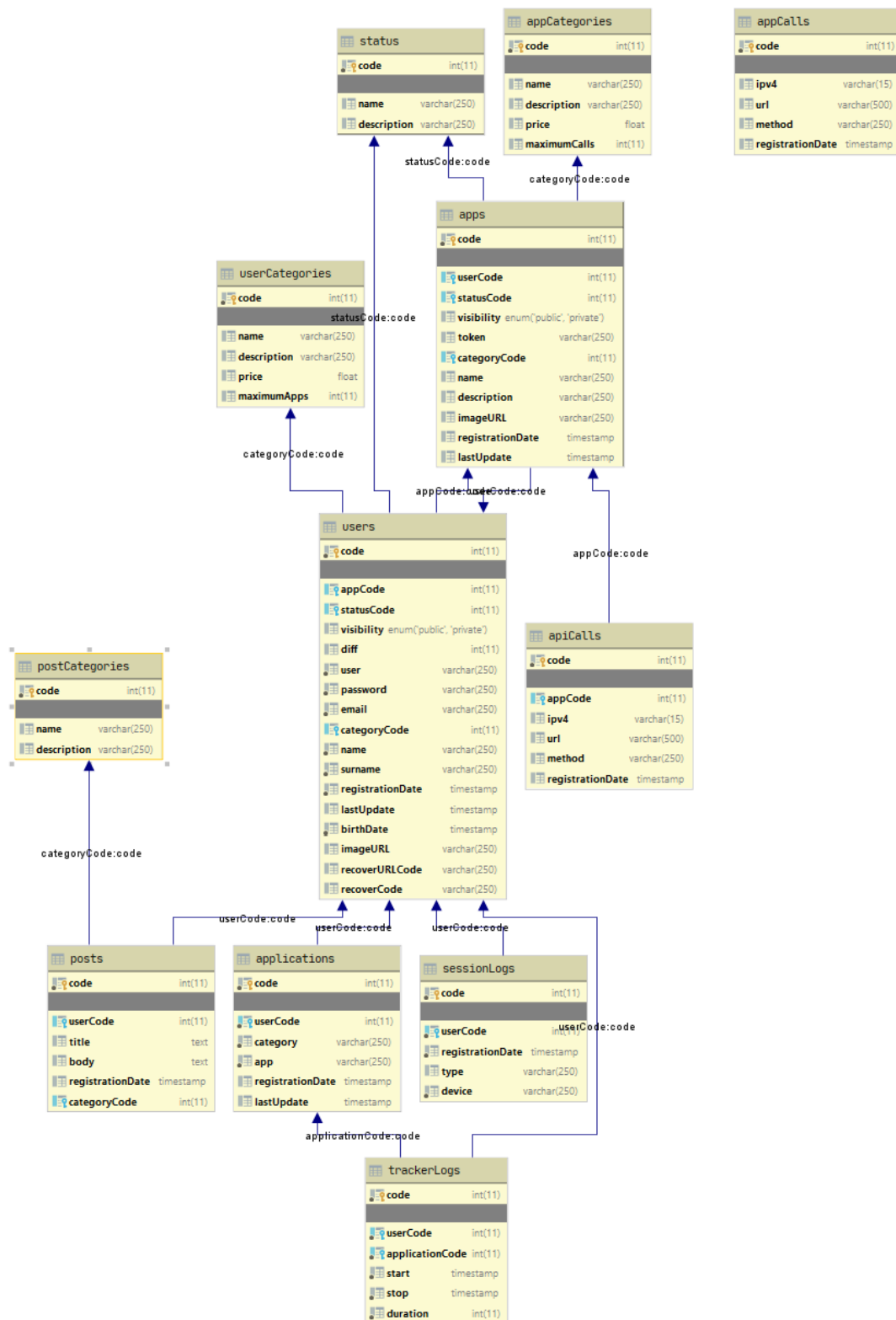
Una vez ya tenemos escogido el sistema de base de datos para nuestra aplicación tenemos que ver que tablas tenemos que crear, que datos almacenará cada una de ellas y las relaciones entre estas.

⁵ <https://www.eversql.com/mariadb-vs-mysql/>

Modelo entidad relación



Modelo relacional



Análisis

En las imágenes anteriores se puede ver el modelo relacional y modelo entidad relación de nuestra base de datos. En ellas podemos apreciar la existencia de 12 tablas, así como la existencia de relaciones entre ellas, todas de tipo 1-N.

Analizando un poco las tablas tenemos una que destaca en especial la tabla de usuarios. Esta tabla como se puede esperar contendrá todos los usuarios del sistema. De estos usuarios tenemos que almacenar información como **code** siendo clave primaria de la tabla, almacenamos el código de la aplicación que creó el usuario **appCode**, **statusCode** que determinará si un usuario puede hacer uso o no de los servicios, **visibility** que indicará si otros usuarios pueden ver ese perfil así como los datos relacionados con este, **diff**, **user** que vendría a ser el nick del usuario, **password** sería la contraseña la cual estará encriptada con la función hash bcrypt, **email**, **categoryCode** que sería el código de usuario ya que nuestra aplicación contendrá diferentes categorías como administrador o usuarios premium, **name** que vendría a ser el nombre de usuario, **surname** apellidos del usuario, **registrationDate** fecha de registro, **lastUpdate** que vendría a ser la fecha de la última modificación del usuario así como el día de su cumpleaños almacenado en **birthDate** así como la URL de la imagen de perfil del usuario **imageURL**. Aparte también tiene un conjunto formado por **recoverURL** y **recoverURLCode** que serían los campos que se usarían cuando el usuario quisiese recuperar su cuenta porque olvidó de la contraseña.

Esta tabla está en relación con **userCategories** que contiene las categorías de los usuarios de las cuales almacenamos código de categoría el cual al igual que el code de la tabla de usuarios sería la clave primaria de la tabla aparte almacenamos un nombre de categoría y descripción, así como su precio y las apps máximas que puede tener creadas.

Luego tenemos que también está en relación con la tabla de apps, esta tabla contiene todas las apps que se crearon. Esta tabla contiene al igual que toda las anteriores un código de la app que es la clave primaria de la tabla, así como el código del usuario que creó la aplicación y el cual tiene control sobre ella también se almacena el token el cual le permite a la aplicación hacer uso de la API aparte del nombre y descripción de la app. También contiene el **statusCode** el cual determina si esa app puede hacer uso de los servicios proporcionados por el sistema, **visibility** el cual determina si otros usuarios tienen acceso a ver los datos de la app, **categoryCode** el cual determinará la categoría de

la app. Ya para terminar la tabla contendrá la URL de la imagen de la app que estará almacenada en el campo ***imageURL***, así como la fecha de registro y última actualización de la app. Luego tenemos que esta tiene una relación con la tabla de ***appCategories*** esto es porque las apps pertenecen a una categoría que les permiten hacer ciertas acciones, así como un sistema que controla las peticiones máximas que pueden hacer permitiendo así monetizar el uso de la API.

La tabla de ***appCategories*** contiene un código de categoría, nombre de la categoría, descripción y precio aparte del número máximo de peticiones diarias que puede hacer la app a nuestra API.

En relación con esta tabla tenemos la tabla de ***apiCalls***, esta tabla lleva un registro de las peticiones que hace cada app a la API permitiendo llevar un control más preciso del sistema. Esta contiene un código que sería la clave primaria de la tabla que sería el número con el cual se registró la llamada, ***appCode*** que sería el código de la app que hizo la llamada, así como la URL solicitada, método empleado, así como la IP de la máquina que realizó la petición y fecha en la cual se realizó.

Aparte de estas tablas tenemos otras que si bien son algo más sencillas son las más importantes ya que la aplicación se basa en registrar las aplicaciones que usa el usuario en su equipo, así como cuanto tiempo y todos estos datos se almacenan en la tabla de ***trackerLogs***.

Esta tabla contiene un código de registro siendo este la clave primaria de la tabla, así como el código de usuario al que pertenece cada registro y el código de la aplicación registrada luego tenemos que también almacena la fecha de inicio de la aplicación esto equivale a cuando el usuario empezó a usar la aplicación en su equipo, así como la fecha cuando dejó de usarla bien porque el cerró o porque cambió de aplicación. Como diferencia entre estos 2 campos tenemos el tiempo en segundos que usó cada aplicación el usuario.

Hemos visto que la tabla de trackerLogs tiene una relación con la tabla de ***applications*** esta tabla contiene un registro con las aplicaciones que los usuarios han usado así la categoría a la que pertenecen cada una. Más en profundidad contiene al igual que todas las tablas del sistema un campo code que sería la clave primaria de la tabla un campo ***userCode*** para relacionar cada aplicación con el usuario al que pertenece aparte de los campos ***category*** y ***app*** que indican a qué categoría equivale cada aplicación así como el

nombre de la aplicación aparte y como medida de control tenemos que hay un campo **registrationDate** que nos proporciona la fecha cuando fue registrada esa aplicación en el sistema así como el campo **lastUpdate** que registra la fecha de la última modificación.

Luego y si bien no cumple una función extremadamente necesaria tenemos la última tabla **sessionLogs** esta tabla nos proporciona un registro de todos los inicios de sesión y cierre de sesión en el sistema, así como el dispositivo desde donde se realizó la acción, el usuario que la realizó gracias a su código de usuario, así como la fecha cuando se realizó la acción.

Como hemos visto en la tabla de usuarios y app estas contienen un campo **status**, este campo indica al sistema si estos recursos están activados y tienen acceso al sistema. Para llevar un mejor control de los status estos están en relación con la tabla **status** mediante un código de estatus. Esta tabla contiene un campo code que sería el campo que conforma la clave primaria, así como los campos **name** y **description** que proporcionan el nombre del estado, así como una pequeña descripción.

La tabla **appCalls** si bien no es la más útil sí que nos permite llevar a cabo un control de quien accede a nuestra página web, si bien está haciendo uso de sistemas de seguimiento y control como Google Analytics esta tabla también nos permite obtener alguna información extra por nuestra cuenta. Esta tabla está conformada por los siguientes campos: **code** que sería el código de petición y clave primaria de la tabla, **IP** de la máquina desde la cual se realizó la petición, URL solicitada, **method** que nos indicará el método de petición empleado, así como la fecha en cual se realizó.

Como última adición durante el desarrollo tenemos **posts**, Estos anuncios están almacenados en la base de datos en la tabla **posts**. Esta tabla es formada al igual que todas por un campo **code** que vendría a ser la clave primaria de tabla, así como por **userCode** que almacenará el código del usuario creador del post, así como los campos **body** y **title** que vendría a ser el contenido y el título de la noticia. Ya para terminar tendríamos un campo fecha que almacenará el momento del registro, así como el código de categoría al que pertenece y que está en relación con la tabla de **postCateogries**.

Esta sería la última tabla de nuestra base de datos y almacena las categorías a las que puede pertenecer un post. Esta tabla está formada por un código de categoría siendo este la clave primaria de la tabla y por los campos **name** y **description** que proporcionan el nombre de la categoría, así como una breve descripción de la misma.

Hemos visto a qué equivale cada campo de las diferentes tablas de nuestra base de datos, pero no hemos visto los tipos de datos que estos usan. Como resumen tenemos que los datos de tipo fecha hacen uso de timestamp (yyyy-MM-dd HH:mm:ss) que nos permiten almacenar tanto el día como la hora aparte en la mayoría de casos donde se hace uso de este tipo de datos tiene un valor por defecto de current_timestamp.

Para los campos code de las tablas los cuales son los campos que forman la clave primaria tenemos que estos son de tipo int con un máximo de 11 dígitos, así como no permiten nulos y por defecto son auto incrementales.

Para los campos de precios hacemos uso de float como dato ya que nos permite tener una gran precisión con números decimales.

Para los campos de tipo texto como nombres, descripciones y demás haremos uso de varchar con un tamaño máximo de 250.

Mockups

Hoy en día es muy importante el diseño de una aplicación ya que, aunque la aplicación funcione muy bien si el diseño no resultaba adecuado a las condiciones donde se va usar la aplicación, usuarios que van a usarla, etc. no valdrá de nada. Para solucionar este problema haremos uso de las recomendaciones que nos proporcionan numerosos profesionales y medios

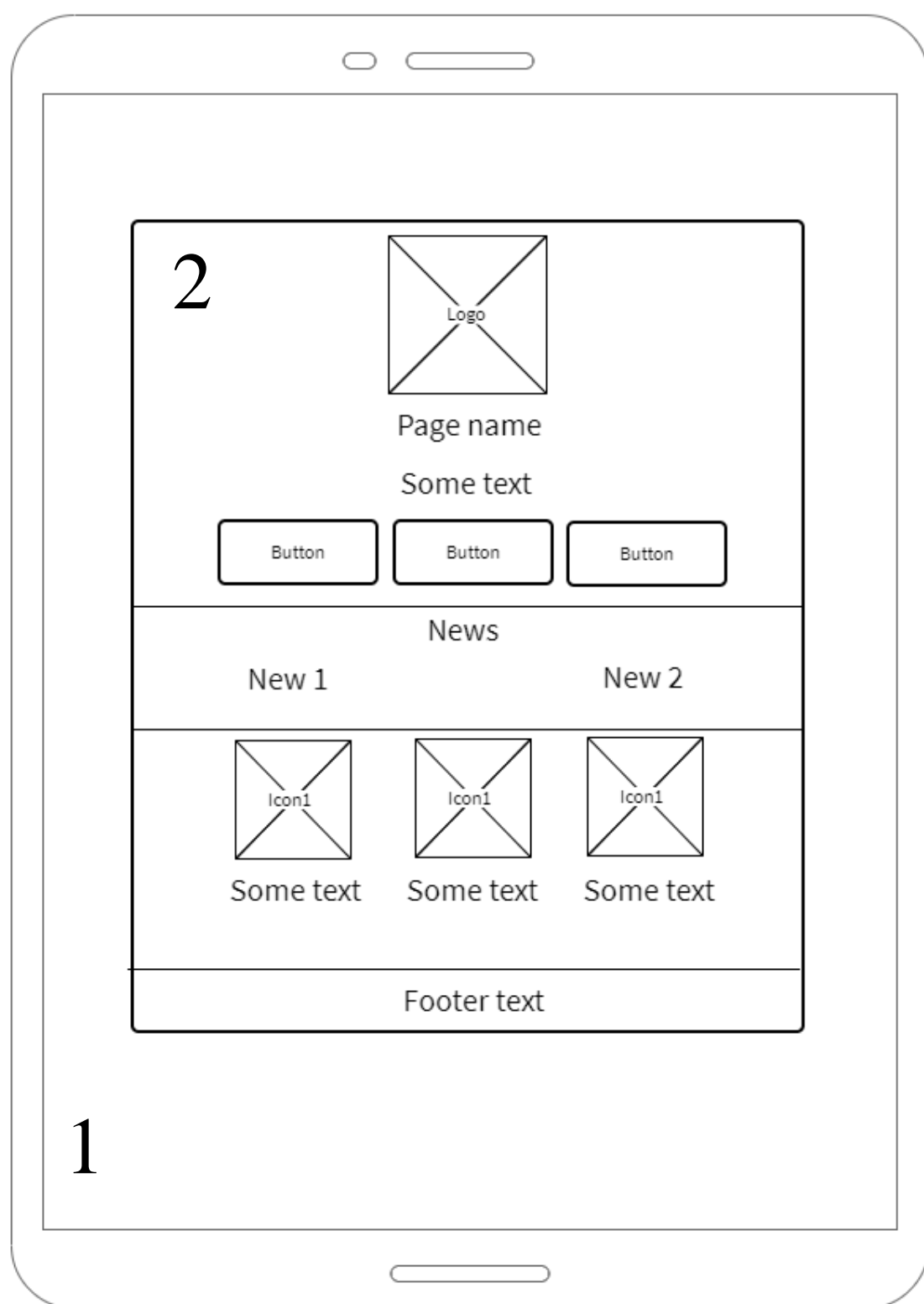
- <https://es.wikipedia.org/wiki/Mockup>
- <https://ingenieriadesoftware.es/generacion-interfaces-usuario-desde-mockups-inteligencia-artificial/>
- <http://blog.inteligencia.com/2017/10/requisitos-mediante-mockups.html>

Los cuales nos recomiendan realizar una serie de diseños preliminar a la codificación, muy simplificados que nos permitan visualizar la posición de los componentes en nuestra aplicación, estos se denominan mockups.

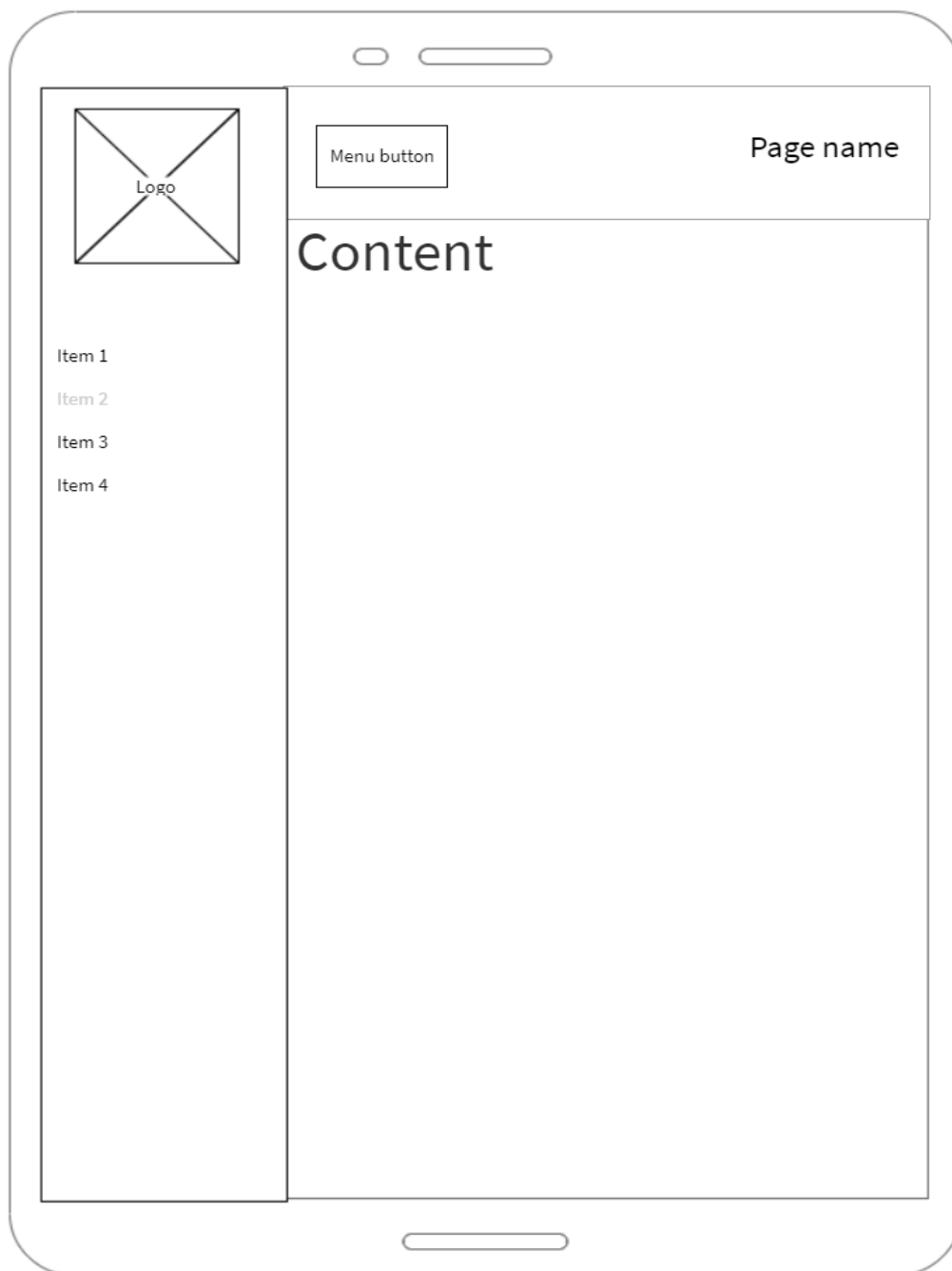
Los diseños que se muestran están realizados teniendo como base un dispositivo móvil ya que según las recomendaciones actuales tenemos que priorizar el desarrollo de las interfaces para estos dispositivos a su vez ya que haremos uso de Bootstrap estos diseños están pensados para ser iguales tanto para la versión móvil como para la versión de navegador de escritorio por lo que únicamente se mostrarán las versiones móviles ya que las únicas diferencias que pueden existir residen en la posición de los diferentes elementos.

Diseño general

Todas las páginas de nuestro sitio web que no requieran de un inicio de sesión es decir a las que todo el mundo sean usuarios registrados o no tienen acceso mantienen un conjunto de elementos y características comunes entre ellas. Estas son que el contenido de los diferentes sitios este contenido dentro de un div (2) el cual a su vez está comprendido dentro de otro div más grande y sobre el cual se mostrara el fondo que serán los colores de la página (1). Dentro del div (2) que contendrá las diferentes opciones, formularios, textos y demás para las diferentes localizaciones de nuestro sitio también tenemos elementos comunes como son el uso del logo de la aplicación, así como el nombre como cabecera para el bloque.

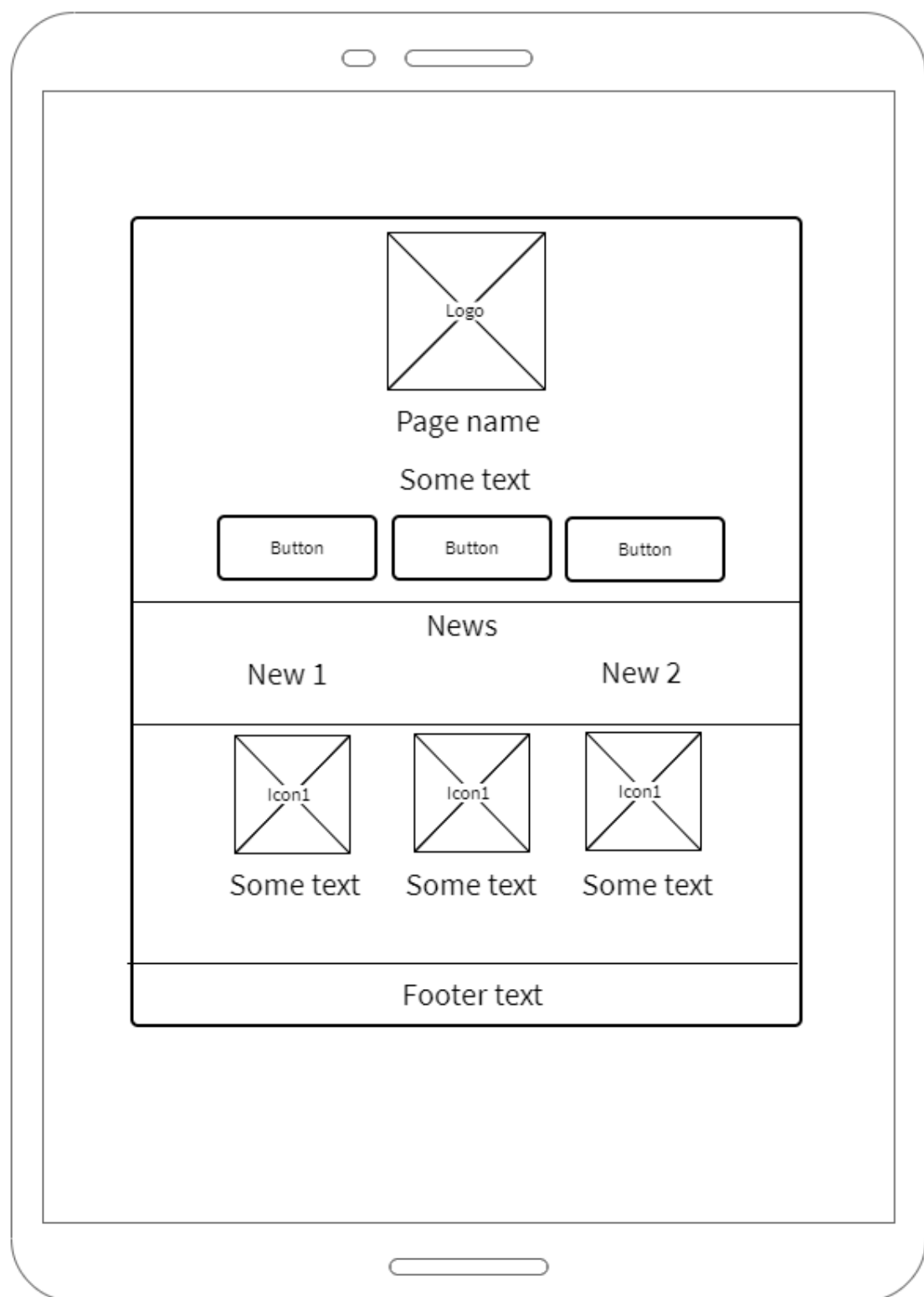


En cuanto a las páginas que requieren de un inicio de sesión tenemos que todas tienen una barra de navegación la cual muestra el botón que permite abrir y cerrar el menú lateral que se encontrara situado la izquierda de la página, así como el nombre de la página. En lo que abarca al menú tenemos que este contendrá el logo de la página, así como un conjunto de submenús que permitirán al usuario recorrer nuestro sitio web. Este menú se adaptará al tamaño de la página. Dentro de estas páginas los diseños que se mostrarán estarán contenido dentro de un bloque del bloque de contenido manteniendo estáticos la barra de navegación como el menú lateral.



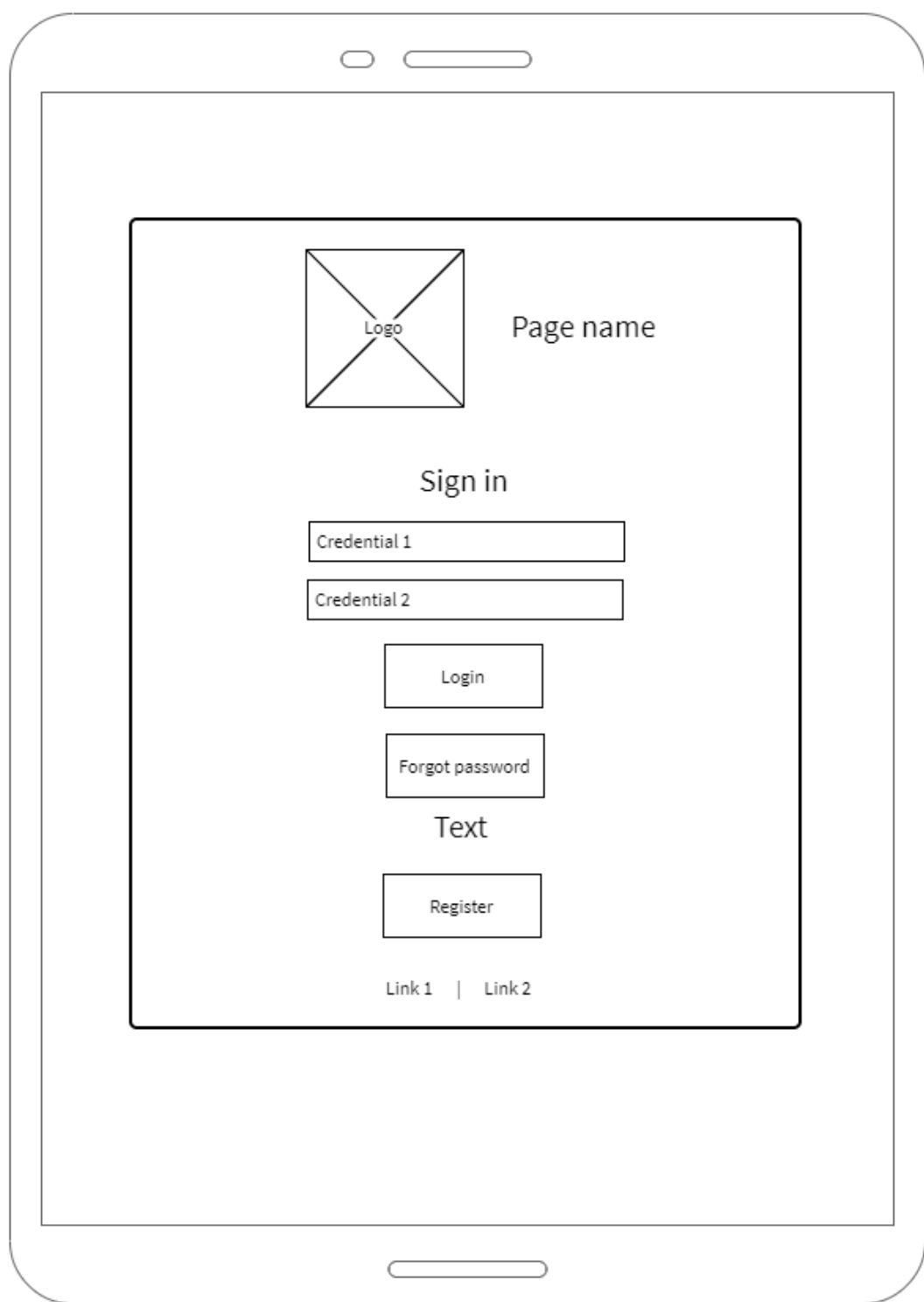
Index

Aquí se muestra la página por defecto de la aplicación cuando accedemos desde un navegador a la URL. Dentro del bloque central vemos que se mostrará el logo de la aplicación, así como el nombre y un slogan. Justo debajo de estos elementos tenemos que se muestran una serie de botones los cuales se tiene pensado que enlacen con la documentación de aplicación, así como con la página de login que veremos más adelante. Justo debajo tendremos una sección de novedades donde se mostrarán las últimas noticias de nuestra aplicación en forma de pequeño resumen. Justo debajo tendríamos unos iconos que representan la esencia de la aplicación, así como un footer que incluiría información extra sobre el autor y licencia.



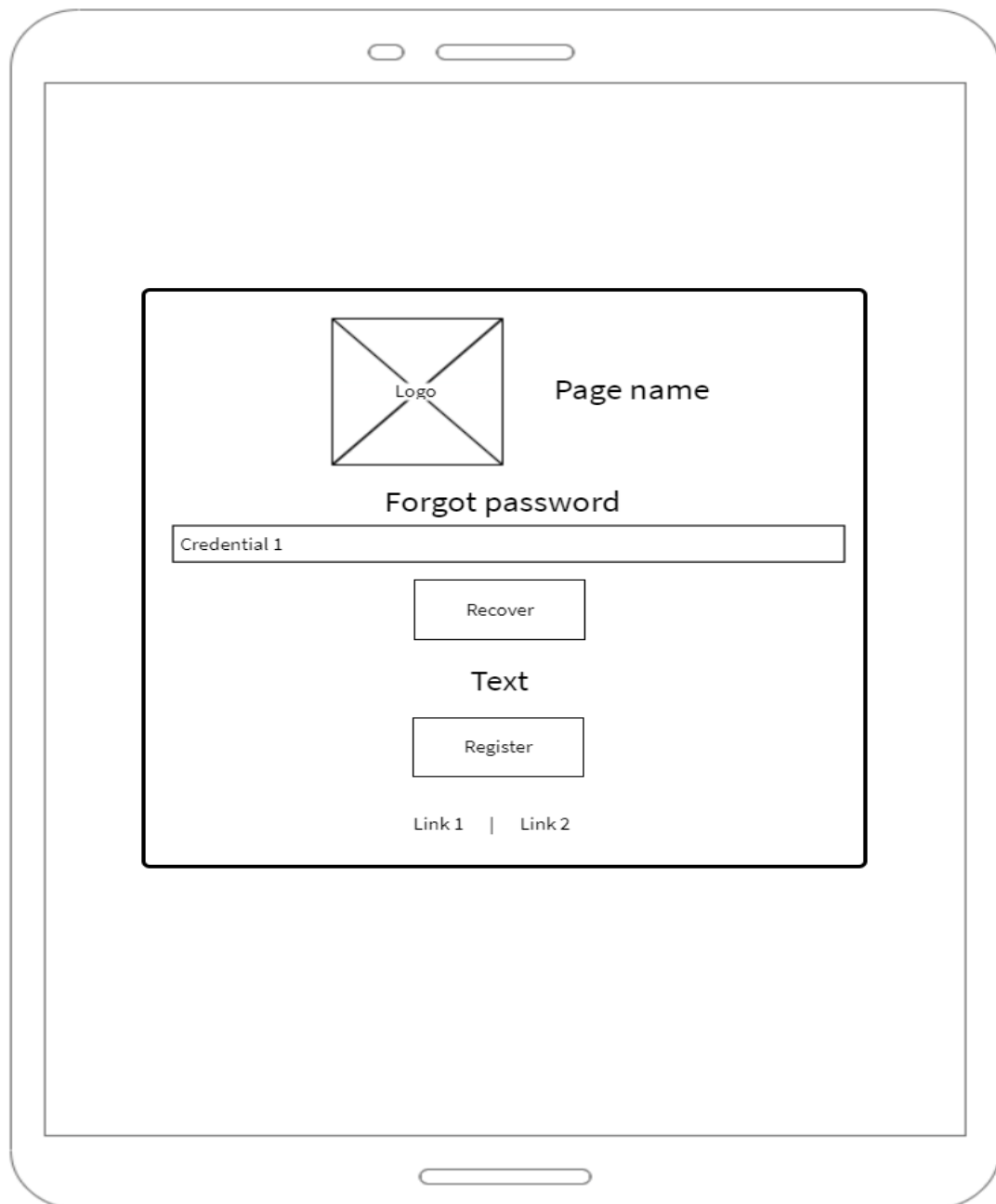
Inicio de sesión (Login)

En este diseño vamos a mostrar la página que los usuarios verán cuando intenten iniciar sesión en nuestra aplicación. Manteniendo la coherencia entre las diferentes páginas del sitio esta también tiene un div con el contenido de la aplicación sobre otro div que contendrá en fondo. En este bloque principal tenemos a su vez y al igual que los otros lugares del sitio el logo de la aplicación, así como el nombre de la misma. Justo debajo tendríamos el formulario de inicio de sesión con las credenciales que se decidan implementar, así como el botón de subir del formulario. A continuación, tendríamos el botón que nos mostraría la página para recuperar la contraseña en caso de que no nos acordemos, así como otro enlace a la página para registrarse en caso de que no se tenga cuenta. Justo debajo tenemos 2 links que en un principio se tiene pensado que muestren la política de privacidad, así como los términos y condiciones que aceptan los usuarios al hacer uso de nuestros servicios. En caso de que ocurra algún error se mostrar una alerta indicando el error que se produjo.



Recuperación de cuenta 1 (forgot)

Vamos a revisar la página de que permite a los usuarios recuperar su cuenta. Este diseño es el más básico de todos los que dispone nuestra página ya que solo incluye a parte de lo típico que es el bloque de contenido con el logo y nombre de la aplicación sobre un div que contiene el fondo, un formulario con un campo y un botón de subir, así como los links a la política de privacidad y términos y condiciones y un botón para acceder al formulario para crear una cuenta.



Recuperación de cuenta 2 (recover)

Este diseño será el que disfrutará el usuario cuando reinicie la contraseña de su cuenta en caso de que la olvidara. En este disponemos de un formulario con 4 campos que serán las credenciales que se requerirán para reiniciar la contraseña de la cuenta, así como un botón de envío de formulario para llevar a cabo la acción. Aparte de estos elementos tenemos los ya típicos de la página y que están explicados en modelos anteriores.

The image shows a wireframe of a mobile application screen for account recovery. The screen is framed by a rounded rectangle representing the device. Inside, there is a central rectangular area containing the form elements. At the top of this area is a square placeholder for a logo with an 'X' and the text 'Logo' below it. To the right of the logo is the text 'Page name'. Below these is the title 'Recover'. The form consists of four input fields: 'Credential 1' (a single wide field), 'Credential 2' and 'Credential 3' (two side-by-side fields), and 'Credential 4' (a single wide field). Below the input fields is a rectangular button labeled 'Recover'. At the bottom of the form area are two links, 'Link 1' and 'Link 2', separated by a vertical bar.

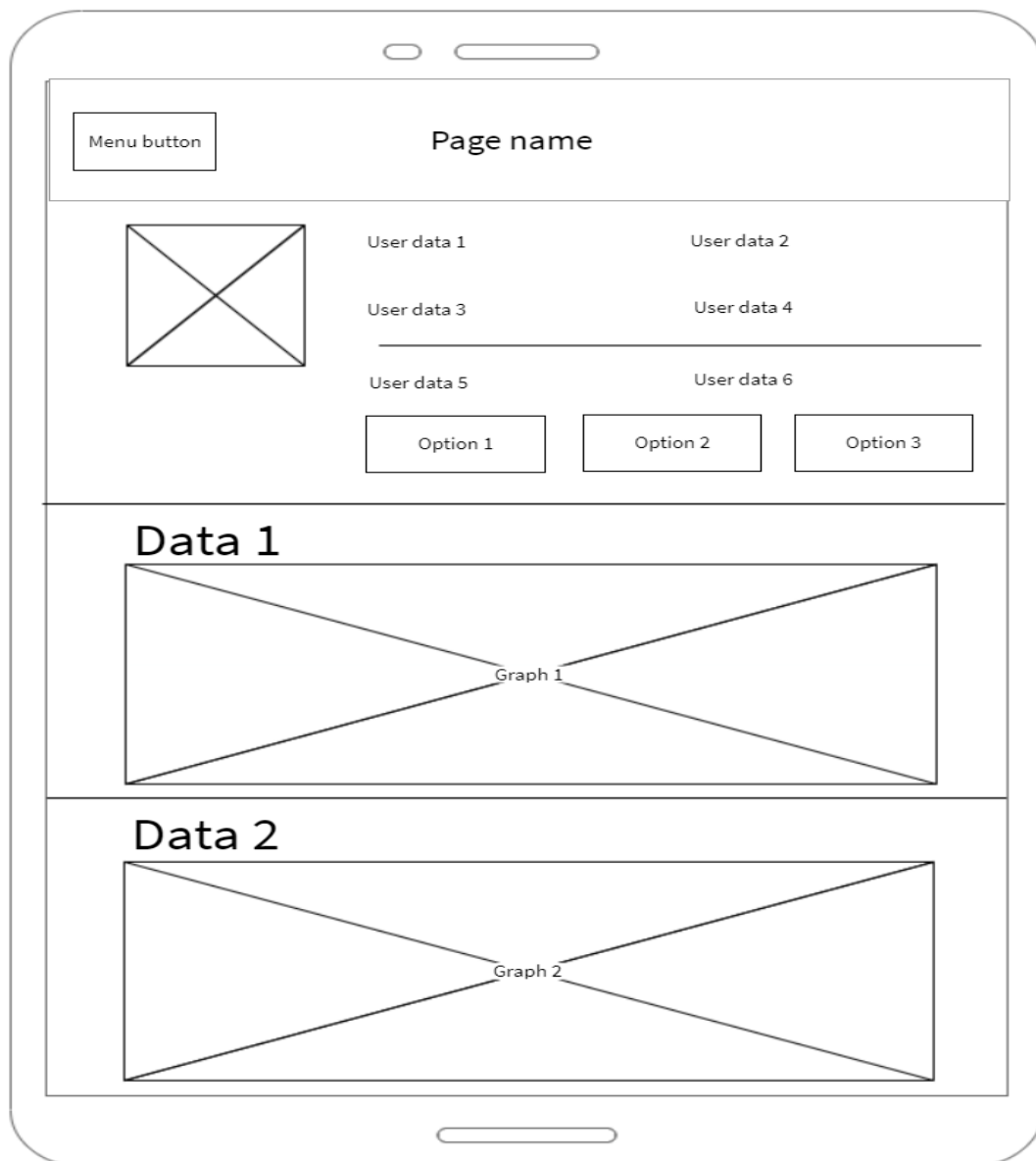
Creación de cuenta

Aquí veremos el diseño que tendrá la página que permitirá al usuario crear una cuenta en el sistema. En este diseño y teniendo como base el diseño típico de nuestro sitio tenemos un formulario con el conjunto de campos a cubrir que serán necesarios para crear el usuario, así como el botón que permite el envío de este. Justo debajo tendríamos un texto explicativo de la siguiente opción que el un inicio es la de ir a la página de inicio de sesión para ello disponemos de un botón. Justo debajo de este tendríamos el conjunto de link que harán referencia a la política de privacidad, así como los términos y condiciones de uso.

The image shows a wireframe of a 'Create account' page displayed on a tablet. The page layout includes a header area with a 'Logo' placeholder and a 'Page name' label. Below this is the main title 'Create account'. The form consists of several input fields: 'Credential 1' (a single wide field), 'Credential 2' and 'Credential 3' (two side-by-side fields), 'Credential 4' (a single wide field), 'Credential 5' and 'Credential 6' (two side-by-side fields), and 'Credential 7' (a single wide field). Below the input fields is a 'Create' button. Underneath the button is a 'Text' label, followed by a 'Login' button. At the bottom of the form are two links, 'Link 1' and 'Link 2', separated by a vertical bar. The entire form is enclosed in a rounded rectangle, which is itself centered within a larger rounded rectangle representing the tablet screen.

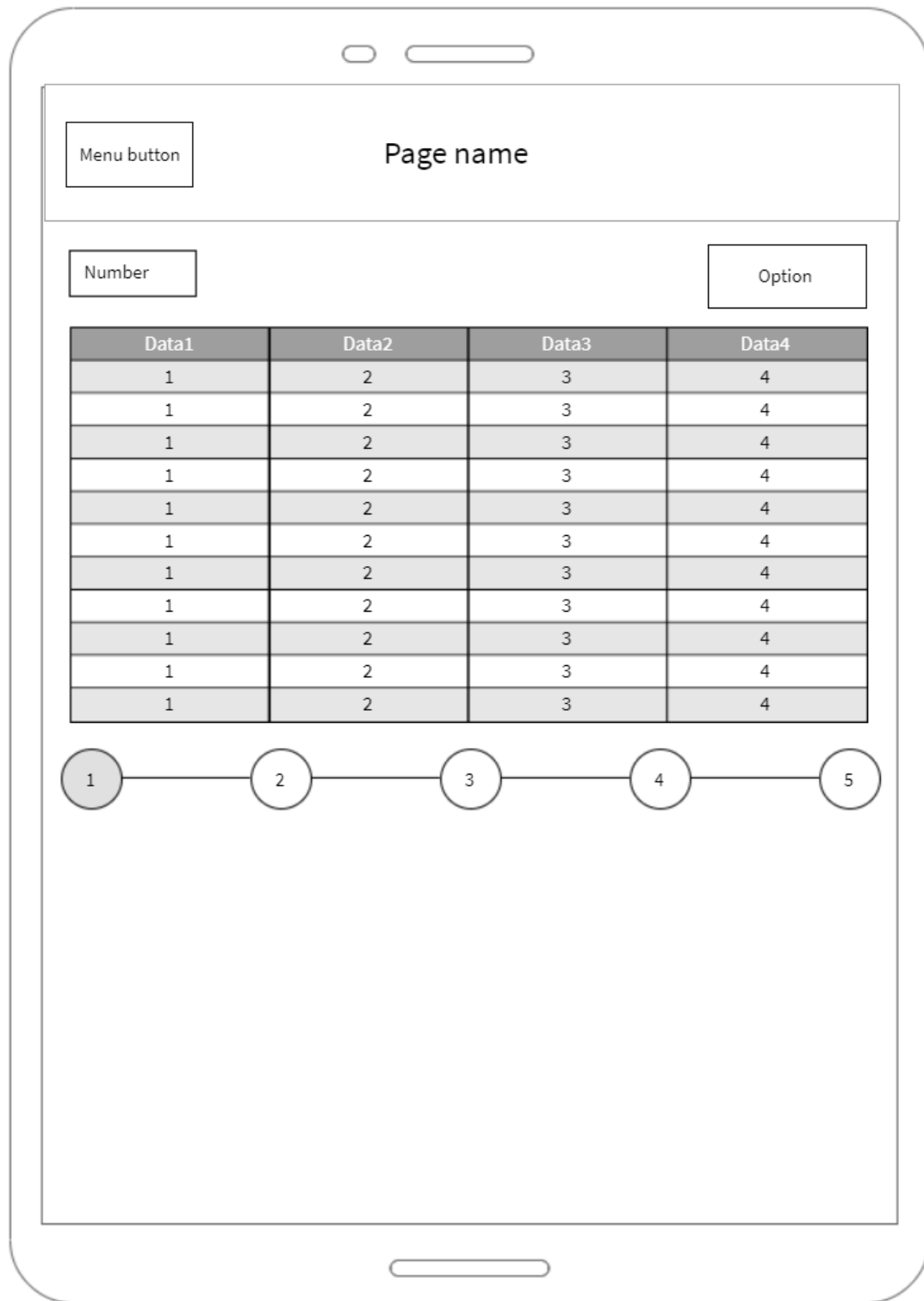
Perfil o App

Esta vista contiene lo que verá el usuario cuando acceda a su perfil o al de cualquier otro usuario o app. En este diseño podemos ver que dentro de la página del perfil se verá una foto del recurso, así como los datos del mismo como nombre, fecha de registro cumpleaños y demás a su vez justo debajo de los datos estarían las opciones que podría hacer dar de baja el recurso o editarlo. Justo debajo tenemos un conjunto de gráficos que mostrarán diferentes datos del usuario o app como tiempo de uso, registros, tiempo diario y demás.



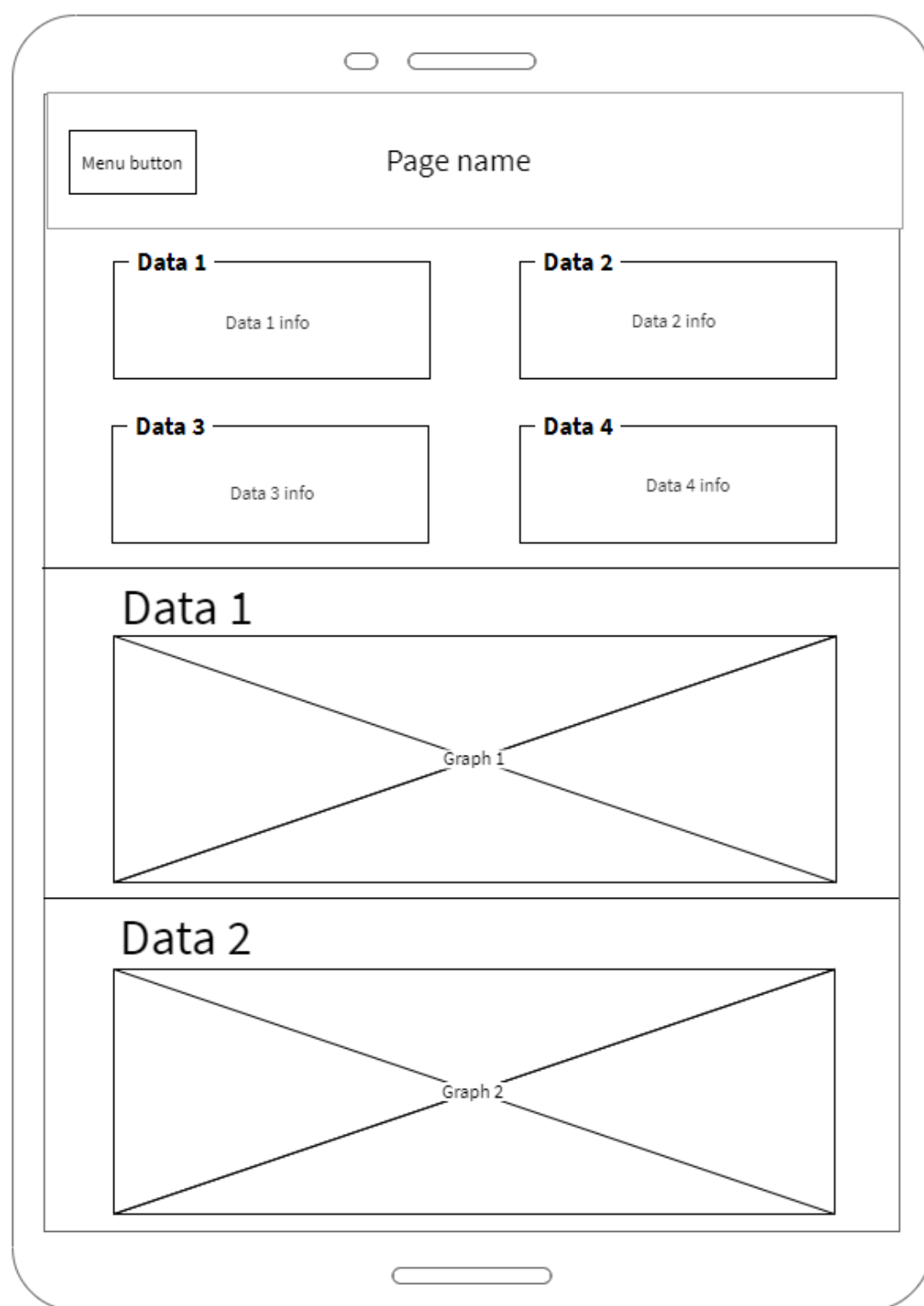
Data 1

Los diferentes datos que muestre la aplicación a partir de los datos recopilados se tienen que mostrar de alguna manera. Para solucionar este problema en un principio haremos uso de estos 2 diseños que se muestran a continuación. El primero sería el empleado para ver datos de una manera masiva como por ejemplo el registro de aplicaciones o de logs. En estos casos haríamos uso de tablas que nos permitirían mostrar todos los registros. Para dar facilidades a estas vistas las tablas tendrán un tamaño personalizable (selector situado en la parte superior izquierda) así como un sistema de paginación u búsqueda por campo que permitirá al usuario visualizar los datos de una manera más cómodo. Además de estas medidas se pueden organizar las tablas según los datos de las cabeceras. Luego estas vistas proporcionarán a los usuarios diferentes funcionalidades como la opción de borrar, añadir o editar los diferentes registros. Para estas acciones se hará uso de un pop que ocupará la pantalla y donde se permitirá al usuario la introducción de datos, así como la confirmación de la acción.



Data 2

Esta es otra vista que usaremos para mostrar los datos a los usuarios. Esta vista es una mucho más trabajada que mostrará datos en forma de gráficos, tanto de líneas, barras, pociones, etc. así como mediante números y porcentajes que resumirá los datos de manera que resulte sencilla de ver. En la primera sección justo después de la barra de navegación tenemos un conjunto de grupos (cards) que contendrán datos en forma de resumen de uso por ejemplo el número de aplicaciones del usuario o el número de logs que ha realizado el día actual entre otros muchos. A continuación, en otro bloque tenemos los gráficos estos darán información mucho más extendida en el tiempo y proporcionarán al usuario la capacidad de interactuar con ella de una manera más sencilla pudiendo ajustar las gráficas cambiar los colores y demás. Estos gráficos además y con el fin de resultar más amigables se podrán ocultar pensando en las pequeñas pantallas de los dispositivos móviles.



Codificación y pruebas

Codificación

Durante el desarrollo del software se ha empleado el sistema de control de versiones GIT⁶. Este sistema nos permite llevar a cabo un control sobre los cambios realizados en los ficheros del proyecto. Dados los múltiples cambios que se realizaron en las fases iniciales del proyecto estos primeros registros incluyen una gran cantidad de cambios por lo que una descripción más extensa de cada uno de ellos sería imposible.

Como resumen del proceso de desarrollo tenemos que inicialmente se desarrolló la parte de API que permite la obtención, modificación y eliminación de los recursos almacenados en la base de datos, así como los mecanismos de seguridad y filtrado. A continuación, se desarrolló la parte del cliente sin registrar es decir la parte de documentación, inicio de sesión, registro y demás. Después de esta primera parte nos faltaba la parte más extensa de todo el proyecto la parte del panel de control donde el usuario una vez registrado e iniciada su sesión tendría acceso a las funcionalidades y datos disponibles. Para el desarrollo de esta parte se fueron desarrollando las diferentes secciones una a una empezando por la página de los usuarios tanto para el usuario, dueño de alguna app o el administrador del sistema. A continuación, siguieron las páginas para los otros recursos como logs, apps o aplicaciones. Cuando ya se tenían las diferentes páginas de la aplicación funcionando correctamente se pensó cómo mejorarlas, así como resultado se produjo la adición de un sistema de filtrado y paginación, así como la posibilidad de mirar apps y perfiles ajenos al nuestro. Estas últimas modificaciones nos obligaron a realizar serios cambios sobre los recursos que se almacenan para adaptarlos a las nuevas funcionalidades permitiendo a los usuarios hacer su perfil público o privado, comprobación de campos para el filtrado, etc. Ya para terminar el proceso de desarrollo se nos ocurrió que ya que la aplicación había cogido toques sociales también estaría interesante tener un sistema de noticias y novedades apareciendo así los posts. Una vez la parte de cliente estaba terminada se subió toda al repositorio remoto y desde entonces los commits son más pequeños e incorporan más explicación de los cambios realizados.

⁶ <https://github.com/ppabli/awesometracker-server/commits/master>

Pruebas

Una vez terminado el desarrollo toca la parte de pruebas y test. Para comprobar que tanto la aplicación web, escritorio como la API funcionan perfectamente se han realizado un conjunto de pruebas que van desde pruebas de caja blanca al testear la introducción de valores incorrectos, nulos u omisión de valores necesarios, así como pruebas de caja negra mediante pruebas que simulaban el uso por parte del usuario final de la aplicación. Ya como última prueba se ha dado a miembros de la familia y amigos acceso al software para que pudieran usarlo y notificar de posibles fallos o mejoras. Las pruebas que si han sido fructíferas fueron las de caja negra ya que nos permitieron detectar fallos en la revisión de los campos y sus valores en el envío de datos, así como las pruebas de uso por parte de usuarios reales que nos permitieron obtener ideas de cómo mejorar la usabilidad del servicio.

Aparte de estas pruebas antes mencionadas también se han realizado pruebas de seguridad que garanticen el cumplimiento de las leyes, así como la seguridad de los datos sensibles de los usuarios.

Futuras mejoras

Dada la limitada cantidad de horas que teníamos disponibles para el desarrollo del proyecto hemos tenido que seleccionar que caracterizas y funcionalidades queríamos tener disponibles para el momento de su presentación, así como las que queríamos implementar en un futuro. Estas futuras mejoras van desde cambios en el diseño hasta introducción de nuevas características y funcionalidades de cara al usuario acompañadas a su vez como no podía ser de otra forma de mejoras de rendimiento, corrección de errores y perfección de la documentación. Las nuevas incorporaciones se pueden agrupar en las siguientes categorías:

Diseño

Una de las mejoras que nos gustaría tener implementado en la versión final es el uso de temas en la aplicación de navegador. Es características del diseño vendría a proporcionar al usuario la capacidad de cambiar los colores de la aplicación eligiendo entre un conjunto de paleta de colores disponibles algo similar al modo oscuro o claro que otras aplicaciones soportan, pero permitiendo una mayor personalización.

Funcionalidades

Uno de los últimos recursos que se han desplegado en la aplicación han sido los posts. Estos nacieron para dar a la aplicación un toque más social y permitir a los usuarios una mayor percepción de los cambios y mejora que presente el sistema. Dado el escaso tiempo restante de cara a la presentación, así como la falta de base para poder realizar esto el uso de los posts se ha limitado a los usuarios con acceso a la base de datos. De cara al futuro nos gustaría eliminar esta limitación y permitir a los usuarios a creación, modificación y eliminación de post, así como la implementación de un nuevo sistema de notificación y sugerencias.

Otra de las funcionalidades que se quedan en el tintero en la versión 1.0 del sistema es una mayor información. Durante el desarrollo nos hemos dado cuenta que a pesar de no recopilar datos sensibles de los usuarios ni de las aplicaciones que estos usan somos capaces de obtener una gran cantidad de información y metadatos que pueden resultar de interés de cara a los usuarios finales. Para ello estamos pensando que otros datos podemos mostrarles en futuras versiones del sistema aparte de los ya presentes.

Aplicación de escritorio

Dadas las limitaciones presentes en el proyecto hemos decidido centrar nuestros esfuerzos en el desarrollo de la aplicación web dejando a la aplicación de escritorio con las funcionalidades mínimas necesarias para su uso. Pero eso va a cambiar para ello hemos estado investigando nuevas opciones de desplegar nuestra aplicación para que llegue a la mayor cantidad de usuarios y sistemas posibles así como que esta incorpore nuevas funcionalidades y características como consecuencia se espera que la nueva aplicación de escritorio pase de usar librerías gráficas y módulos de Python y haga uso de tecnologías como ElectronJS unificando así las tecnologías empleadas en el desarrollo de la aplicación de navegador y escritorio.

Bibliografía

A continuación, se listan las páginas más empleadas durante el desarrollo.

- <https://cloud.google.com/>
- <https://www.opencodez.com/javascript/build-restful-api-using-node-js.htm>
- <https://github.com/>
- <https://auth0.com/learn/token-based-authentication-made-easy/>
- <https://vuejs.org/v2/guide/>
- <https://www.w3schools.com/>
- <https://developer.mozilla.org/es/>
- <https://stackoverflow.com/>
- <https://www.youtube.com/>
- <https://mariadb.com/kb/en/documentation/>
- <https://manuais.iessanclemente.net/index.php/Portada>

Anexos

Anexo 1

Enlace a la documentación presente en la aplicación web:

<https://awesometracker.ddns.net/docs>

Anexo 2

Enlace a la aplicación web:

<https://awesometracker.ddns.net>

Anexo 3

Enlace a los repositorios de GitHub:

<https://github.com/ppabli/awesometracker-server>

<https://github.com/ppabli/awesometracker-desktop>

Anexo 4

Enlace a la lista de reproducción con los videos donde se muestra el funcionamiento del servicio:

<https://www.youtube.com/playlist?list=PL-1Ze9GVKLwzRJGEltgkMI-ICTp4Jo66a>