

Propuesta de acuerdo del proyecto

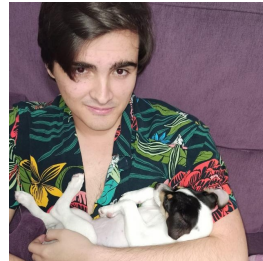
Ingeniería web

Grupo 13

Octubre 2021



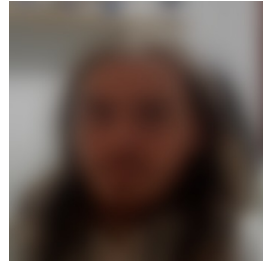
Líder: Salcedo Navarro, Andoni (785649)



Pelayo Benedet, Tomás (779691)



Subías Rodríguez, Rubén (759406)



Velasco Calvo, Isaac (758986)

Objetivos

Se han analizado las características del equipo de trabajo y los requerimientos que son necesarios para afrontar el proyecto y se ha llegado a la conclusión de que los objetivos del equipo de cara al proyecto son los siguientes:

- O1. Ser capaz de identificar las necesidades para desarrollar una aplicación Web que se pueda desplegar en producción.
- O2. Ser capaz de adaptar las nuevas funcionalidades propuestas y desplegarlas en el sistemas web ya funcional.
- O3. Comprender cómo funcionan las pasarelas publicitarias para poder monetizar la aplicación.
- O4. Comprensión de las arquitectura detrás de los canales de comunicación multicast.
- O5. Entender cómo gestionar el almacenamiento y encriptado de datos sensibles.
- O6. Aprender a utilizar los conceptos de UX/UI para que la usabilidad de la aplicación sea la adecuada para el usuario final.

Funcionalidades

- F1. La aplicación implementará un chat grupal en el que se podrán compartir enlaces con otras personas que usen la aplicación (O1,O2,O4). Con el objetivo de aprender cómo implementar canales de comunicación bidireccionales.
- F2. La aplicación soportará un inicio de sesión (O1,O2,O5,O6). Con el objetivo de que cada usuario pueda acceder a sus uri 's creadas para poder modificarlas, compartirlas, añadir, eliminar, monetizar, con la finalidad de entender cómo se realiza el control de sesión y gestión de contraseñas en un nuevo framework.
- F3. Se incluye un sistema de redirección para que el usuario pueda monetizar el número de clicks sobre sus usuarios (O1,O3). Con el objetivo de entender cómo se crea un modelo de negocio real a partir de una aplicación.
- F4. La aplicación verificará que la uri que el usuario quiere crear es alcanzable (O2,O5). El objetivo es aprender cómo se puede comprobar si una redirección es válida, para evitar redirecciones dañinas o inexistentes.

ID	Peso	Funcionalidad	Objetivos
F1	15	La aplicación implementará un chat grupal en el que se podrán compartir enlaces con otras personas que usen la aplicación.	O1, O2, O4
F2	10	La aplicación soportará un inicio de sesión.	O1, O2, O5, O6
F3	25	Se incluye un sistema de redirección para que el usuario pueda monetizar el número de clicks sobre sus uri's.	O1, O3
F4	10	La aplicación verificará que la uri que el usuario quiere crear es alcanzable.	O2, O5

Cuadro 1: Tabla de reparto de pesos.

Orientaciones sobre la descripción de funcionalidades

F1	La aplicación implementará un chat grupal en el que se podrán compartir enlaces con otras personas que usen la aplicación.
Cumplimiento	<ul style="list-style-type: none">▪ Los usuarios pueden entrar a un chat y en él pueden interactuar entre ellos▪ Todos los usuarios participantes en el chat reciben todos los mensajes▪ Se pueden compartir links de los enlaces recortados de cada usuario en el chat
Escalabilidad	<p>5:</p> <ul style="list-style-type: none">▪ No necesita escalar <p>10:</p> <ul style="list-style-type: none">▪ Cumplor nivel 5▪ El chat dispondrá al usuario emoticonos para poder enviarlos▪ El chat marcará la hora a la que se recibieron los mensajes <p>15:</p> <ul style="list-style-type: none">▪ Cumplor nivel 10▪ El chat hará entender al usuario que su mensaje ha sido mandado con éxito▪ El chat permitirá deslizar la lista de mensajes para poder ver los que se mandaron con anterioridad
Profesionalidad	Si esta funcionalidad se implementa con un API Web, la parte expuesta al usuario está documentado con Open API 3.0 AND El API expuesto al usuario tendrá pruebas automáticas que verificarán el cumplimiento AND El código del API expuesto al usuario estará documentado

F2	La aplicación soportará un inicio de sesión.
Cumplimiento	<ul style="list-style-type: none"> ▪ La información que se necesita se obtiene previamente, por ejemplo, en el momento en que el usuario se identifica. ▪ A la hora realizar una operación, se lanzará un proceso asíncrono que se encargue de actualizar la lista a mostrar. ▪ La lista de URIs se debe devolver en un tiempo máximo configurable.
Escalabilidad	<p>5:</p> <ul style="list-style-type: none"> ▪ No necesita escalar. <p>10:</p> <ul style="list-style-type: none"> ▪ La información que se necesita se obtiene previamente, por ejemplo, en el momento en que el usuario se identifica. ▪ A la hora realizar una operación, se lanzará un proceso asíncrono que se encargue de actualizar la lista a mostrar. ▪ La lista de URIs se debe devolver en un tiempo máximo configurable.
Profesionalidad	<ul style="list-style-type: none"> ▪ El API expuesto al usuario tendrá pruebas automáticas que verificarán el cumplimiento. ▪ Las pruebas se centrarán en la integración de la API, de tal forma que se comprobará el estado de la lista mostrada al usuario tras realizar las distintas operaciones disponibles. ▪ Se documentará apropiadamente el código del API expuesto al usuario.

F3

Se incluye un sistema de redirección para que el usuario pueda monetizar el número de clicks sobre sus URIs

Cumplimiento

- El usuario debe poder crear nuevas URIs, las cuales se guardaran en el servidor Web, basadas en URIs externas y alcanzables
- El usuario crea estas URIs a través de una petición HTTP con el servidor Web.
- Estas nuevas URIs generadas deben de guardarse en el servidor Web, y también se tiene que guardar que usuario la ha generado.
- Opcionalmente, el usuario puede decidir que debe recibir "puntos.^{en} su cuenta por cada vez que una de las URIs generadas por este es usada.
- Otro usuario, al acceder a una de las URIs generadas, será redirigido a una página de espera, donde tendrá que esperar 5 segundos para continuar a la URI sin "shortening".

Escalabilidad

5:

- Un usuario con una petición HTTP GET, puede acceder a una de las URIs generadas/acortadas. Los puntos obtenidos por acceder a las URIs son virtuales y no se pueden canjear por dinero de verdad.

10:

- Cumple el nivel 5
- Una nueva petición de "shortening" de URI, tipo HTTP POST, se ejecuta en un nuevo hilo paralelo, donde este realiza la comprobación de validez de la URI dada por el usuario, la generación de una nueva URI y su guardado en el servidor Web.
- Entrar en la URI acortada crea un nuevo hilo en el servidor Web, el cual, asincrónicamente al usuario, le da un punto a quien creó la URI acortada.

15:

- Cumple el nivel 10
- La petición de "shortening.^{es} gestionada primero por un hilo, el cual añade a una cola, global en el servidor Web, la petición en cuestión.
- Otro hilo distribuye las peticiones de la cola entre una serie de hilos workers, los cuales realizan las funciones de comprobación de validez, generación de URIs y guardado en el servidor Web.
- La página de espera al acceder a una URI acortada contiene anuncios de verdad, los cuales generan dinero.

20:

- Cumple el nivel 15
- La comprobación y el guardado de la URI generada en el servidor Web se realizan de forma asincrónica tras generar la URI y remitírsela al usuario.
- Esta URI no es funcional hasta que los dos procesos anteriores terminen.

25:

- Cumple el nivel 20
- El acceso a una URI acortada y el aumento de puntos que tiene un usuario se realiza también con una arquitectura Master-Worker.
- Aquellos puntos obtenidos por un usuario con sus URIs personales pueden ser canjeados por dinero real con una petición HTTP.

Profesionalidad

- La API de las peticiones HTML de creación de nuevas URIs debe estar documentada.
- Debe haber una serie de pruebas automáticas sobre la API de creación de nuevas URIs.

Cumplimiento

- La aplicación verificará al crear la URL que esta es alcanzable (se puede hacer una petición http GET que en un tiempo razonable devuelve una respuesta 200).
- Si esta funcionalidad se implementa con un API Web su implementación será consistente con la semántica de http
- POST /link comprueba que la URL es alcanzable y si no lo es devuelve una respuesta con un error 400.
- GET /hash nunca podrá redirigir a una URL que antes no se haya comprobado que sea alcanzable. Si el hash está registrado pero la URL de destino no está verificada se devuelve una respuesta con un error 400.
- Si alguna de las dos peticiones anteriores devuelve errores de tipo 400 por más de un motivo deberá devolver un objeto JSON especificando el motivo concreto del error del usuario.

Escalabilidad

5:

- No necesita escalar

10:

- Cumple el nivel 5
- El servidor devuelve inmediatamente la URL acortada de forma condicional ya que su verificación la va a realizar algún proceso asíncrono
- La URL acortada no estará operativa mientras que la URL no sea validada

15:

- Cumple el nivel 10
- El trabajo asíncrono de verificación estará gestionado por una cola de tareas en la que se publicará trabajos
- Una máquina cliente se subscribirá a la cola de tareas para realizar trabajos de verificación
- El resultado de la verificación se devolverá por otra cola de tareas

20:

- El trabajo anterior se realizará por dos máquinas clientes en lugar de una
- Las peticiones de verificación utilizarán librerías de tolerancia a fallos y tendrán limitadas el número de peticiones por segundo
- Se asegurará de alguna manera que si una máquina cliente se cae todas las verificaciones que estaban en proceso se volverán a realizar

25:

- Cumple el nivel 20
- Las máquinas del punto anterior expondrán información sobre su nivel de carga de trabajo
- Una aplicación monitorizará esta información y creará o apagará máquinas de una de forma automática para asegurar que siempre hay una máquina disponible y que solo cuando sea necesario hay dos

Profesionalidad

- La API de las peticiones HTML de creación de nuevas URIs debe estar documentada.
- Debe haber una serie de pruebas automáticas sobre la API de creación de nuevas URIs.