



Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías
Departamento de Ciencias Computacionales

Programación para Internet

Una captura de pantalla de un dispositivo móvil que muestra la interfaz de inicio de sesión de una aplicación llamada 'Cool Video'. La pantalla tiene un fondo gris claro. En la parte superior, hay un icono de una cámara y un estado de la barra superior con los símbolos de Bluetooth, Wi-Fi, señal celular y la hora '10:07 p.m.'. El logo de 'Cool Video' está centrado en la parte superior, consistiendo en un icono de una cámara azul y el texto 'Cool Video' debajo. Debajo del logo, hay un formulario de inicio de sesión con el título 'Iniciar Sesion'. El formulario contiene dos campos de entrada: 'Correo electronico' y 'Contraseña', ambos con el mismo texto de ejemplo. Debajo de los campos, hay un botón azul con el texto 'INICIAR SESION'. En la parte inferior del formulario, hay un enlace azul que dice '¿No Tienes Cuenta? Regístrate'.

Proyecto Final - CoolVideo

Jimenez Rodriguez Mario
González López Kevin Daniel
2018 – B

El proyecto por realizar consistía en crear una página web responsiva, además, empaquetar la misma con algún framework y así hacer posible el instalar la página como una aplicación en la plataforma deseada, ya sea Android o iOS.

Los principales objetivos por calificar en el proyecto son los siguientes:

- **Tab-bar o navigation-bar:** Se solicita que la Web App tenga una barra de navegación fija en la parte inferior de la misma, la cual permite una navegación fácil y siempre disponible a las diferentes pestañas de la web.
- **Lista dinámica de elementos:** La Web App debe ser capaz de mostrar siempre la información actualizada, es decir, siempre tiene que estar mostrando lo que actualmente se encuentre en la base de datos.
- **Intercambio de datos:** Es importante que, al realizar un cambio en nuestra base de datos, este se vea reflejado directamente en la Web App, por lo cual su funcionamiento debe ser dinámico, obteniendo en tiempo real los datos solicitados.
- **Sistema CRUD:** Debe ser posible el administrar las publicaciones, desde crear, leer, actualizar y eliminar los mismos.
- **Manejo de multimedia en DB:** En alguna parte de nuestra Web App, debe tener la capacidad de postear, mostrar y eliminar archivos multimedia en la misma.
- **Diseño:** La Web App debe tener un diseño trabajado y bien cuidado, es decir, se debe ver bien tanto en una computadora de escritorio como en la aplicación hecha para alguno de nuestros dispositivos.
- **Imágenes:** La aplicación debe ser capaz de trabajar con imágenes, mostrando, eliminando y subiendo las mismas al servidor.
- **Empaquetar como aplicación para cualquier plataforma a elegir:** El proyecto, una vez realizado, se debe empaquetar en alguna plataforma móvil, ya sea iOS o Android, además de que debe quedar funcional de igual manera que la web.
- **Bonus: mapas, video o notificaciones:** Como extra, la Web App debe de integrar como bonus mapas de Google, videos con su respectivo reproductor interno o el manejo de notificaciones.

Una vez dejado claro los puntos necesarios para la realización de nuestro proyecto, comenzaremos a hablar como se cumplió cada uno de estos, además del mencionar con que se realizó el mismo y como fue el deploy de este.

CoolVideo – Red social de imágenes y videos.

El proyecto que a mi punto de vista cumplía con todos los requisitos de este, fue una red social, en la cual, los usuarios deberán registrarse y verificar su correo para poder tener acceso a la aplicación, una vez dentro de esta, los usuarios podrán ver y realizar publicaciones de imágenes o videos, el cual deberá tener título, una descripción y el contenido multimedia, además, podrán realizar búsquedas de publicaciones, ya sea por título, descripción o usuario que lo público, por último, cada usuario será capaz de administrar sus publicaciones desde la pestaña de **“perfil”**, donde tendrán la posibilidad de ver todas sus publicaciones organizadas por fecha de publicación de manera descendiente, además, podrán acceder al archivo multimedia publicado y podrán eliminar este si así lo desean, cabe mencionar que desde esta misma podrán cerrar sesión si así lo desean.

El proyecto se creó con las siguientes tecnologías:

- Python y Django (Backend)
- Bootstrap
- HTML
- CSS
- jQuery
- JavaScript

Al realizar un proyecto con Django, la principal problema que se presentó, fue el hecho de encontrar un host que soporte Python como interprete, había dos posibilidades, pagar algún hosting el cual lo soporte y que traiga una configuración predefinida o pagar un servidor dedicado en el cual se debería realizar el deploy completo del proyecto desde cero, por reto personal y fines de aprendizaje, tome la decisión de optar por el camino “difícil”, montar mi propio servidor desde un host dedicado, este fue adquirido en DigitalOcean, una vez adquirido y creado el servidor dedicado corriendo el sistema operativo de Ubuntu, se instalaron todas las dependencias, entre las cuales se encuentran Python, virtualenv, MySQL Server, Apache2 y Django.

A continuación, se demostrará a detalle el cómo se cumplen los objetivos del proyecto.

- **Tab-bar o navigation-bar:** Se inserto en la parte inferior de la página, un footer estático en el cual se podrá navegar en las diferentes pestañas de nuestra Web App.

- **Lista dinámica de elementos:** En la página principal de nuestra Web App, se visualizarán siempre las publicaciones más recientes, de igual manera al realizar una búsqueda y al entrar a nuestro perfil, se verán todas las publicaciones realizadas hasta el momento por cada usuario.
- **Intercambio de datos:** Cualquier cambio o publicación que se realice, afectará de manera inmediata en la base de datos, por ende, pasará igual con todos los usuarios que estén usando la misma.
- **Sistema CRUD:** Los usuarios podrán crear y leer publicaciones desde la página principal, además, podrán administrar las mismas desde la página de perfil de usuario.
- **Manejo de multimedia en DB:** Las imágenes podrán visualizarse y administrarse desde la pantalla de inicio, búsqueda y perfil de usuario.
- **Diseño:** La Web App se realizó utilizando Bootstrap, por lo cual, se pudo realizar un proyecto completamente adaptivo para todo tipo de dispositivos.
- **Imágenes:** Como ya se mencionó en puntos anteriores, se podrán subir, actualizar y administrar imágenes desde los posts creados.
- **Empaquetar como aplicación para cualquier plataforma a elegir:** Se empaquetó la Web App con Córdova, utilizando un servicio en línea para facilitar el trabajo sin necesidad de instalar nada, con el simple link de nuestro dominio pudimos realizar este trabajo sin complicarnos con instalaciones extras.
- **Bonus: mapas, video o notificaciones:** Como bonus, se agregó la posibilidad de reproducir y administrar videos en nuestra Web App.

La base de datos fue estructurada de manera sencilla, normalizándola lo más posible para evitar datos recursivos en la misma, creando así solo los campos necesarios para su correcto funcionamiento y mantenimiento.

Tabla Usuario:

```
class Usuario(models.Model):
    email = models.CharField(max_length=50)
    password = models.CharField(max_length=50)
    nombre = models.CharField(max_length=50)
    apellidoP = models.CharField(max_length=50)
    apellidoM = models.CharField(max_length=50)
    codigoVerificacion = models.CharField(max_length=6)
    verificado = models.BooleanField(default=False)
    creado = models.DateTimeField(default=timezone.now())
    editado = models.DateTimeField(default=timezone.now())

    def __str__(self):
        return '{}'.format(self.nombre + ' ' + self.apellidoP + ' ' + self.apellidoM)
```

Tabla Publicación:

```
class Publicacion(models.Model):
    usuario = models.ForeignKey(Usuario, on_delete=models.CASCADE)
    titulo = models.CharField(max_length=30)
    descripcion = models.CharField(max_length=255)
    creado = models.DateTimeField(default=timezone.now())
    editado = models.DateTimeField(default=timezone.now())

    def __str__(self):
        return '{}'.format(self.titulo)
```

Tabla Archivo Publicación:

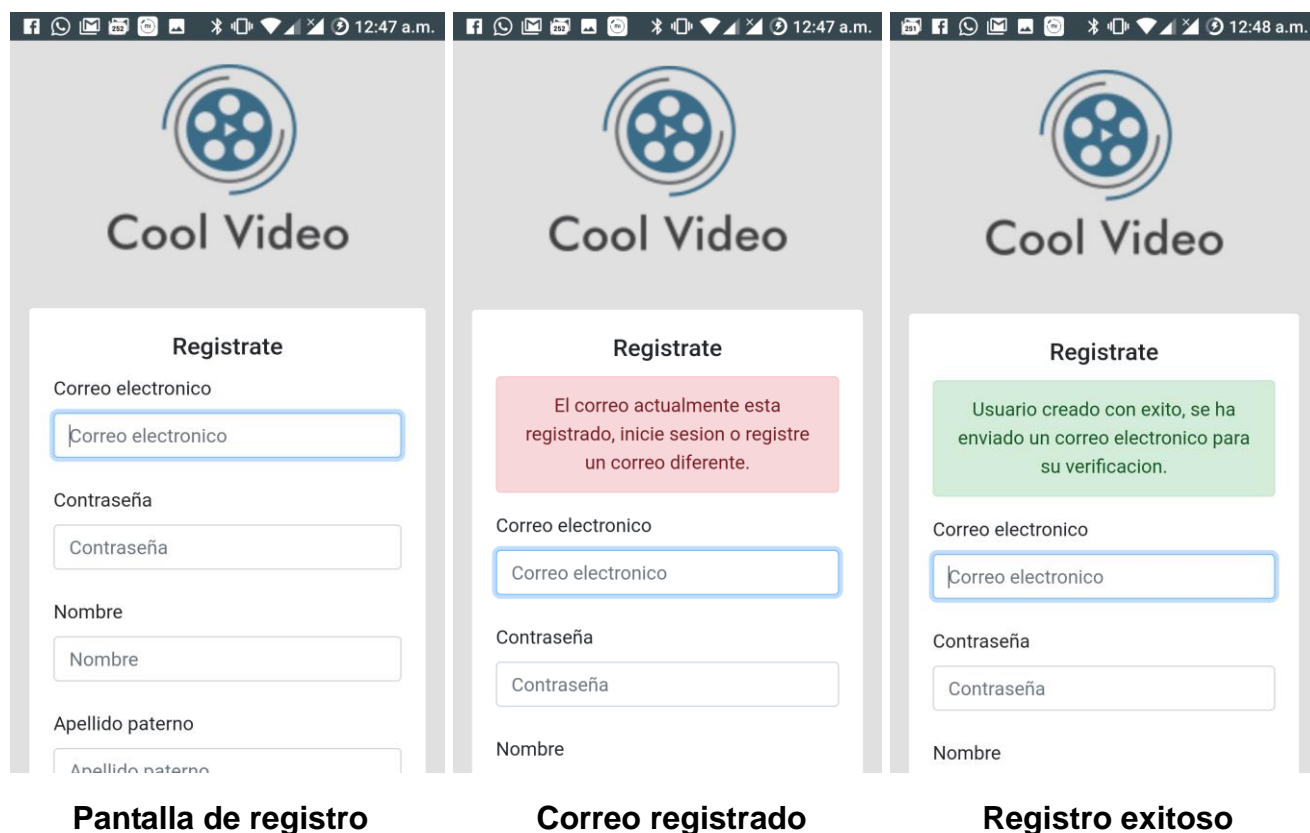
```
class ArchivoPublicacion(models.Model):
    publicacion = models.ForeignKey(Publicacion, on_delete=models.CASCADE)
    archivo = models.FileField(null=True, blank=True)

    def __str__(self):
        return '{}'.format(self.pk)
```

Por último, explicaremos de manera superficial el cómo se realizan las principales funciones de nuestra página web, en las cuales se encuentra el registro, verificación, log-in, crear post, visualizar posts, buscar y administrar, tanto en la interacción del usuario como el trabajo que se realiza desde el lado del servidor.

Pantalla de Registro:

En esta pantalla se solicitan los datos principales del usuario, los cuales son un correo electrónico, una contraseña y su nombre completo, en caso de que el correo introducido en esta pantalla, este regresa un error el cual especifica que el correo ya está registrado y que es necesario registrarse con otro correo o iniciar sesión con el mismo que intento registrarse, una vez completado el registro de manera exitosa, se le especifica al usuario que se envió un correo electrónico en el cual deberá verificar el correo para poder acceder a la página web.

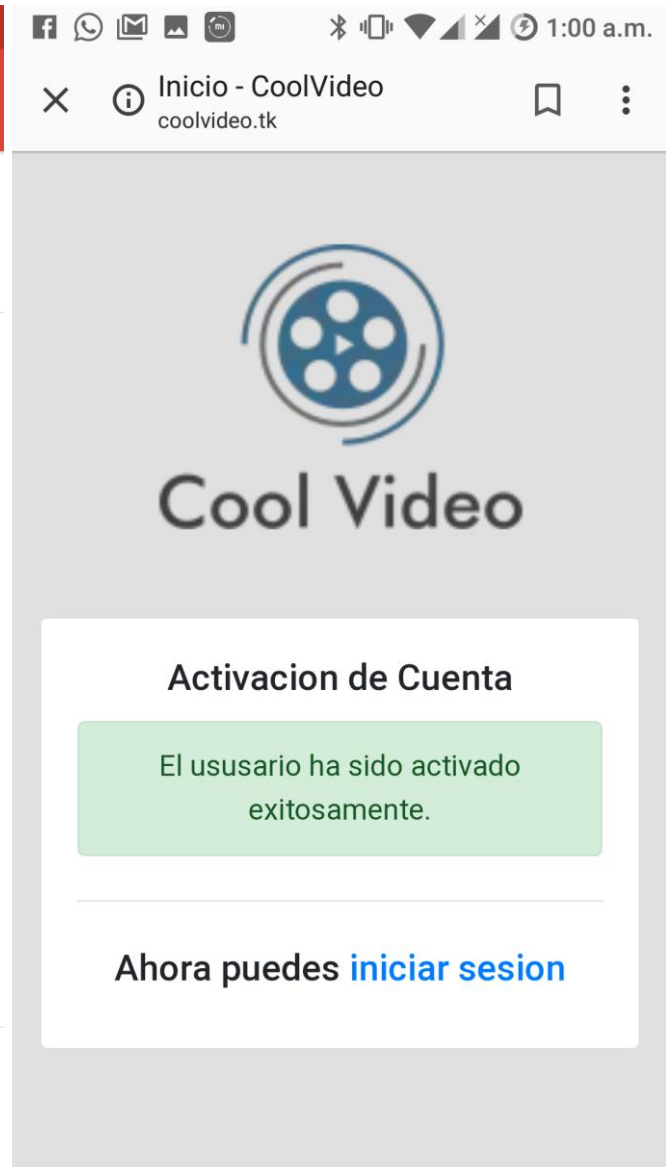


Pantalla de Verificación:

Una vez completado el registro, se enviará un código de verificación a la cuenta de correo en el cual incluirá un URL con el código único generado para poder dar de alta específicamente el usuario registrado al cual se mandó el correo electrónico, una vez que se acceda a dicha página, se activará el usuario y este podrá acceder a la Web App.



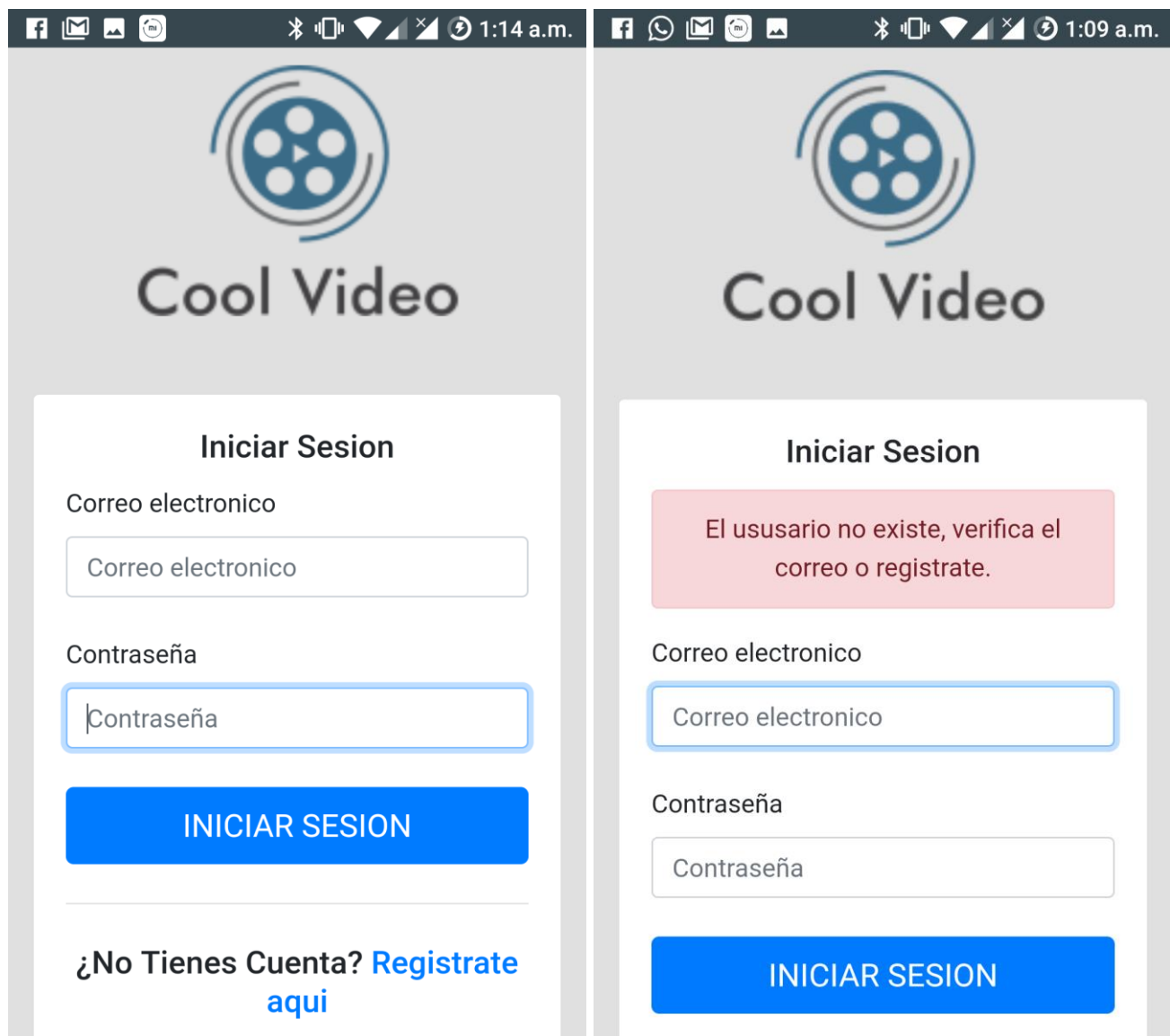
Correo recibido



Cuenta verificada

Pantalla de Inicio de Sesión:

Desde esta pantalla, el usuario podrá acceder a la Web App una vez que haya activado su cuenta, en caso de que el usuario no exista, se le especificara al usuario que ese correo no está registrado y se le invitara a iniciar sesión con algún correo registrado o a registrar el correo con el que intento iniciar sesión, por otro lado, si el correo esta registrado pero la contraseña es incorrecta, se le indicara que verifique el campo ya que el valor ingresado y el registrado no coinciden, en caso de que el usuario este registrado, pero no activado, se le hará mención que antes de poder acceder, deberá activar su cuenta desde el URL que se envió a su correo, por último, en caso de que el usuario este registrado y que su cuenta se encuentre activada, podrá iniciar sesión de manera exitosa y podrá visualizar la página de inicio de la Web App.




The image displays two versions of the 'Cool Video' login interface. Both screens feature the 'Cool Video' logo at the top. The left screen, titled 'Iniciar Sesión', shows a standard login form with fields for 'Correo electronico' and 'Contraseña', a blue 'INICIAR SESION' button, and a link to '¿No Tienes Cuenta? Regístrate aquí'. The right screen, also titled 'Iniciar Sesión', shows the same form but with an error message in a pink box: 'El usuario no existe, verifica el correo o regístrate.' above the email field. Below the screenshots are the labels 'Iniciar sesión' and 'Correo no registrado'.

Iniciar sesión

Correo no registrado

1:11 a.m.



Cool Video

Iniciar Sesión

Contraseña incorrecta, favor de verificarla.


Correo electronico

Contraseña

INICIAR SESION

Contraseña incorrecta

1:13 a.m.



Cool Video

Iniciar Sesión

El usuario no esta verificado, por favor hazlo desde el link enviado a tu correo.

Correo electronico

Contraseña

INICIAR SESION

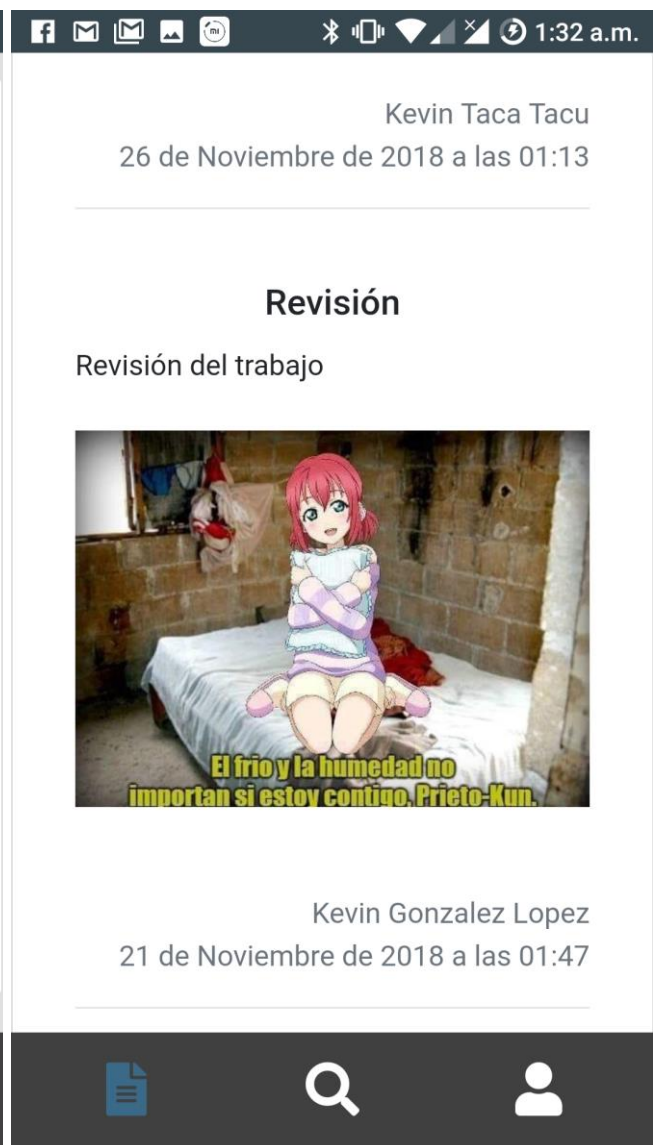
Usuario sin verificar

Pantalla Principal:

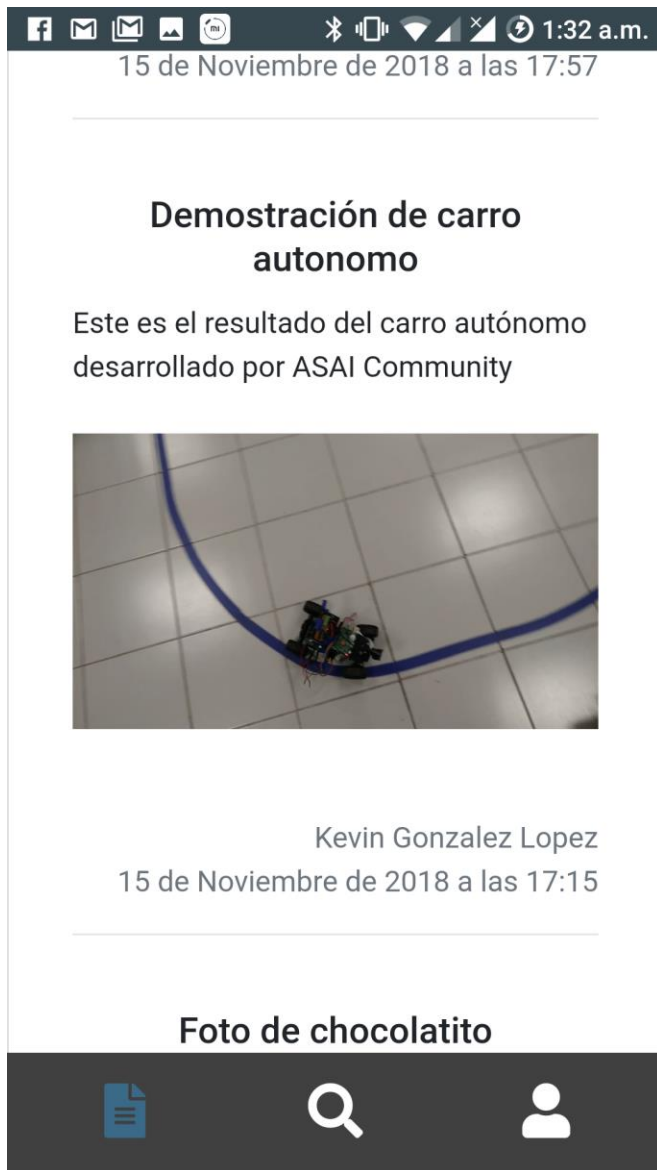
Una vez que el inicio de sesión se haya realizado con éxito, el usuario podrá visualizar la pantalla principal de la Web App, donde de primera instancia se le invita a este el realizar una publicación, solicitando un título, descripción y foto o video para el post, una vez publicado podrá ver de manera inmediata su publicación en la parte de arriba de la misma pantalla, por otra parte, más abajo en esta misma pantalla, el usuario podrá visualizar todas las publicaciones realizadas por los diferentes usuarios participantes de Web App, mostrando el título, descripción, contenido, usuario y fecha de publicación de cada una de las publicaciones realizadas. Cabe mencionar, que una vez estamos en esta pantalla, podremos ver el nav-bar ubicado en la parte inferior de nuestra aplicación de manera fija, la cual nos permitirá navegar en las diferentes pantallas que nos ofrece.



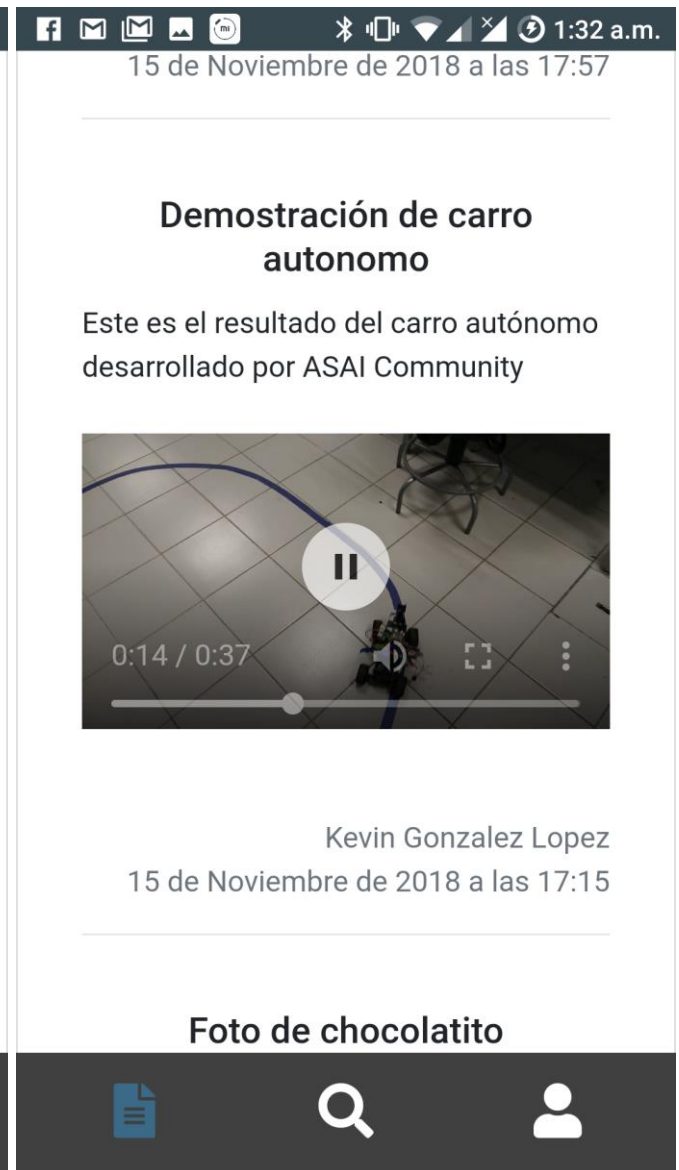
Pantalla principal



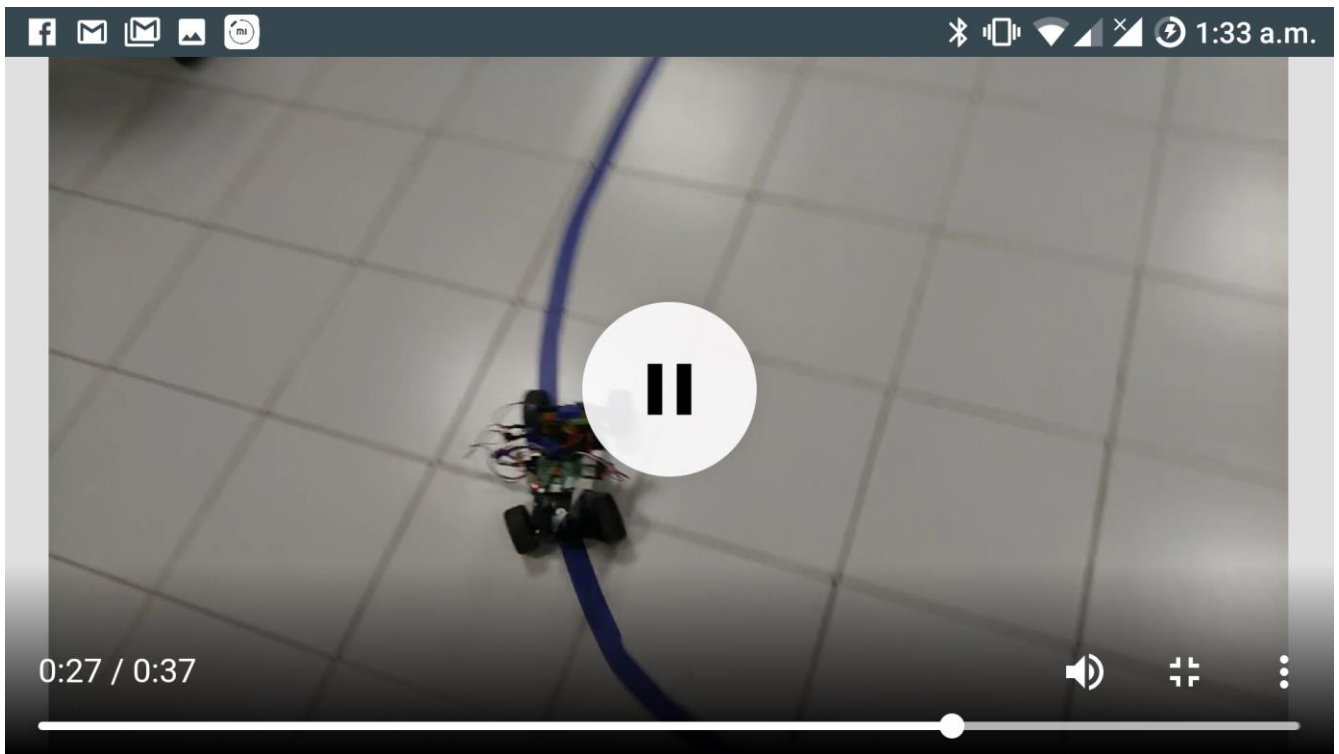
Publicación con imagen



Publicación con video (1)



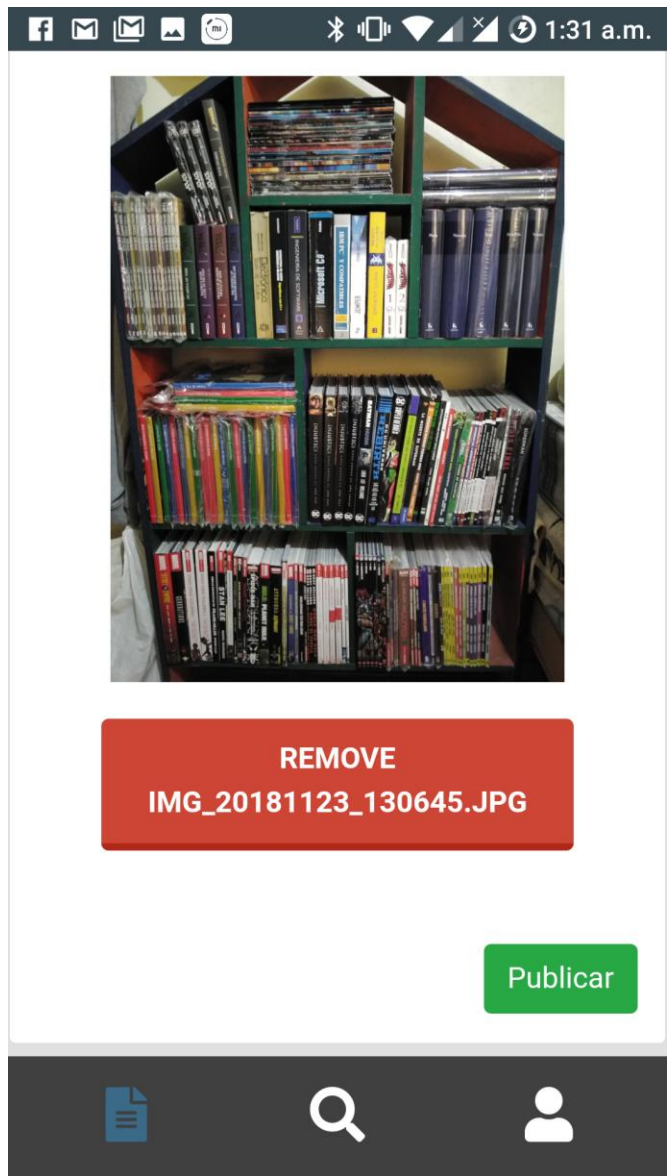
Publicación con video (2)



Visualización de video en pantalla completa



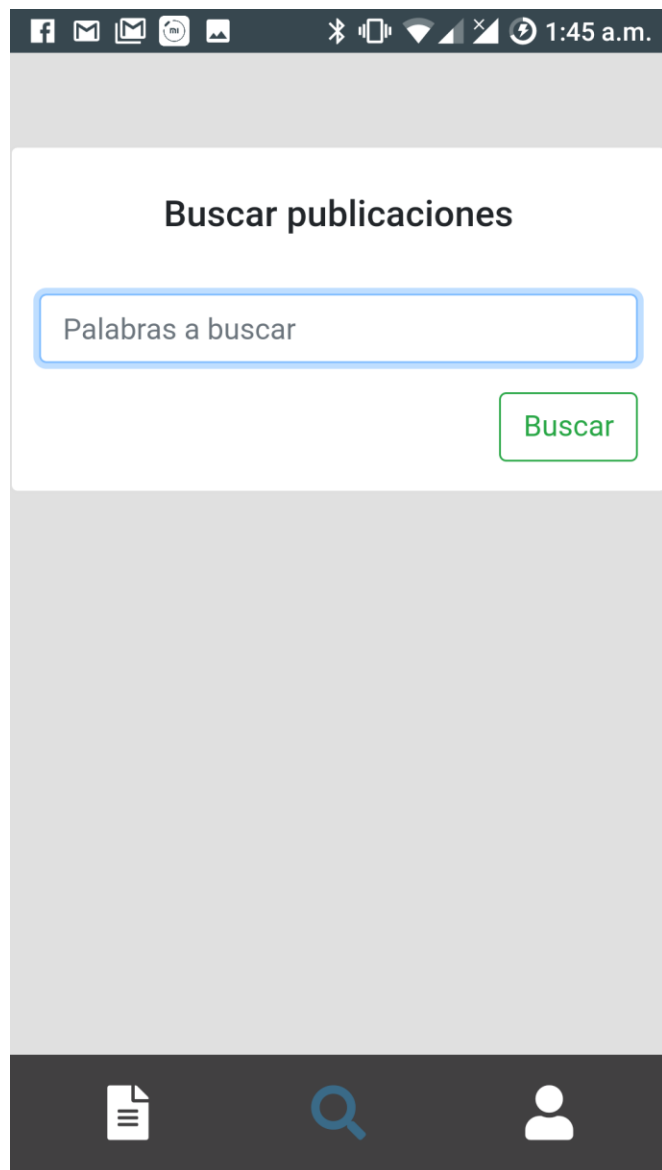
Creación de publicación (1)



Creación de publicación (2)

Pantalla de Búsqueda:

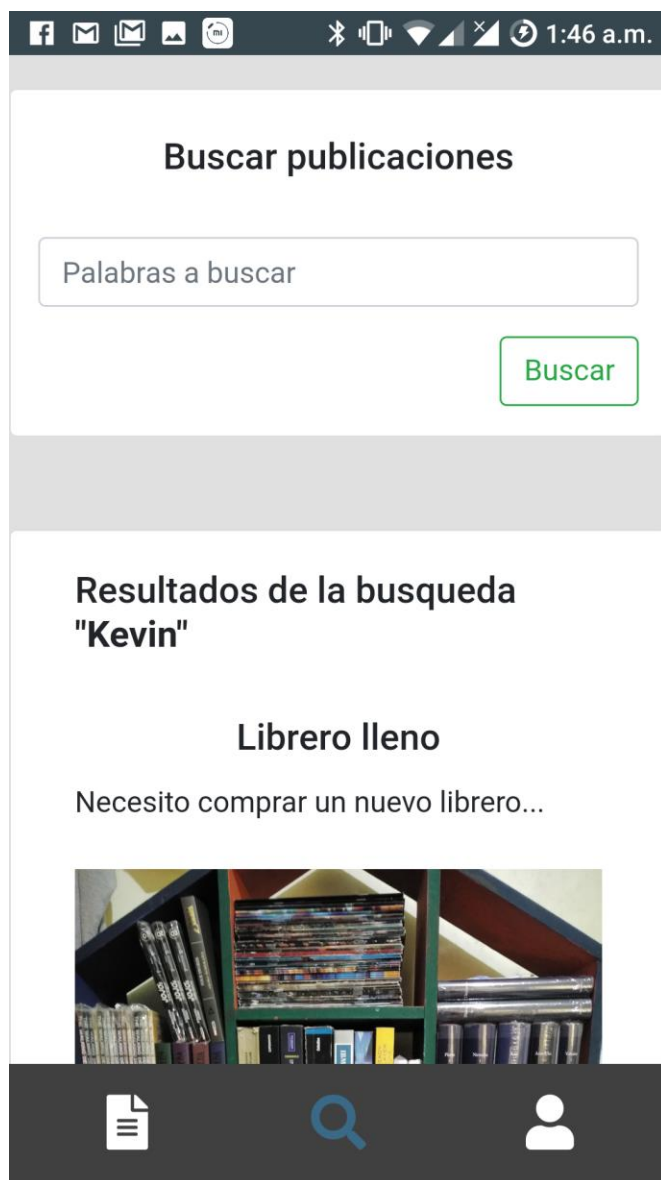
Desde esta pantalla, el usuario podrá realizar búsqueda de cualquier publicación, teniendo la posibilidad de filtrar desde fragmentos de texto, hasta descripciones completas, el buscador es capaz de encontrar coincidencias con el título, la descripción o el nombre completo de algún usuario, en caso de que no se encuentre ningún resultado, el sistema le avisara al usuario que no se encontró nada relacionado a lo que busco, en caso de que la búsqueda tenga éxito, se visualizaran todos los resultados relacionados a la búsqueda.



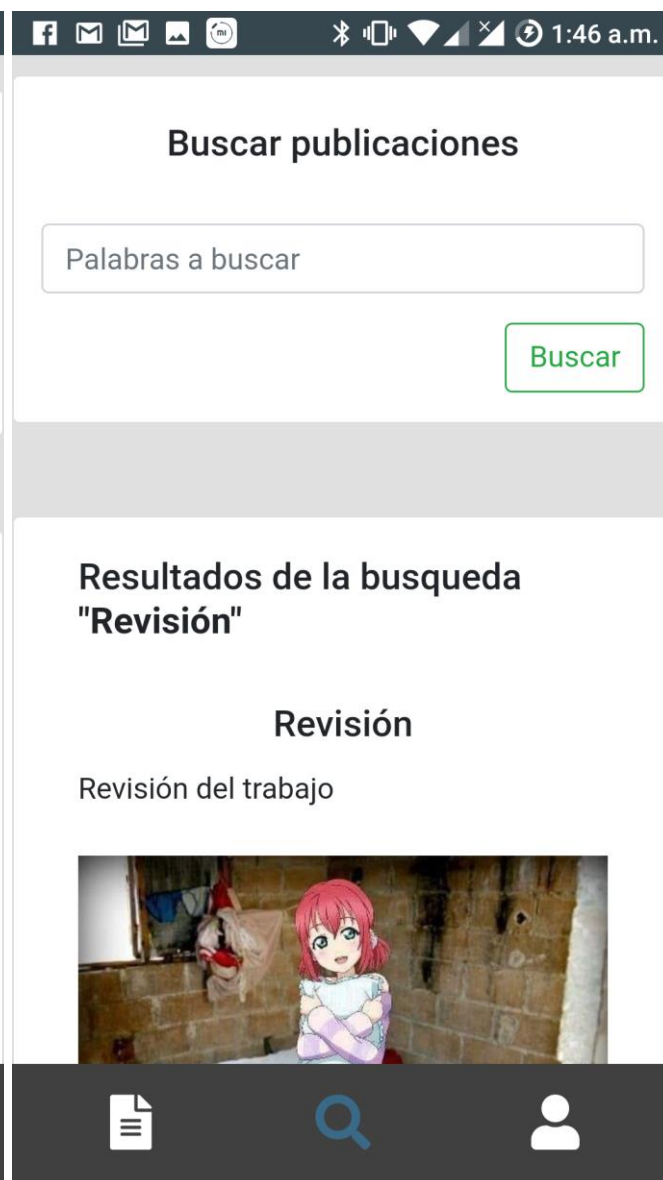
Pantalla de búsqueda



Búsqueda sin resultados



Resultado de búsqueda exitosa (1)



Resultado de búsqueda exitosa (2)

Pantalla Perfil de Usuario:

Desde esta pantalla, el usuario podrá visualizar sus datos personales otorgados, además de esto, podrá administrar sus publicaciones realizadas, las cuales se mostraran en una tabla responsiva con los posts ordenados por fecha más reciente arriba, desde esta tabla podrá visualizar el título, la descripción y fecha de cada una de sus publicaciones, nombre del archivo subido y podrá eliminar cualquiera de estos si así lo desea, de igual manera, podrá cerrar sesión desde esta pantalla si así lo quiere, de ser así, al presionar el botón será redirigido de manera automática a la pantalla de inicio de sesión para que vuelva a acceder con la misma u otra cuenta si así lo prefiere.

Titulo	Descripcion	Publicado
Librero lleno	Necesito comprar un nuevo librero...	26 de Noviembre de 2018 a las 01:13

Perfil de usuario (1)

Archivo	Operaciones
IMG_20181123_130645.jpg	Eliminar

Perfil de usuario (2)

Código del Proyecto:

Todo este proyecto puede ser descargado de manera gratuita en mi cuenta de GitHub personal, este se puede testear de manera local teniendo todas las dependencias instaladas o se puede probar directamente desde la página web en la que se encuentra alojado.

- **Cuenta GitHub:** <https://github.com/DanpezPlay>
- **Link de Repositorio:** <https://github.com/DanpezPlay/CoolVideo>
- **Página Web:** <http://coolvideo.tk/>

Conclusión:

Gracias a este curso, pude aprender el cómo utilizar una de las tecnologías más populares como lo es PHP a la hora de crear páginas web, además de esto, reforcé mis conocimientos de maquetado y diseño de páginas web con los conocimientos dados de HTML y CSS, por último, pero no menos importante, el aprender sobre diferentes frameworks para el empaquetado de nuestra web y así poder convertirla en una Web App generando una aplicación nativa para poder instalarla en los diversos dispositivos fue una práctica bastante interesante y gratificante, ya que aprendimos a como realizar esto sin tener un proyecto creado desde el inicio con este fin, es decir, sin utilizar herramientas de terceros que sirven para realizar esto desde el comienzo. El poder realizar el proyecto final con tecnologías ajenas a lo visto en clase fue lo mejor a mi punto de vista, ya que de esta manera no solo te quedas con lo visto en clase, sino que puedes ampliar aún más tus conocimientos dándonos esta oportunidad.