



# Programación con tecnologías web

## Parcial No. 1

### Instrucciones generales

1. Lea completamente las instrucciones antes de iniciar
2. Suba el proyecto a un repositorio en su cuenta personal en GitHub
3. Genere un release antes del final de la clase, envíe la URL del repositorio via BloqueNeon

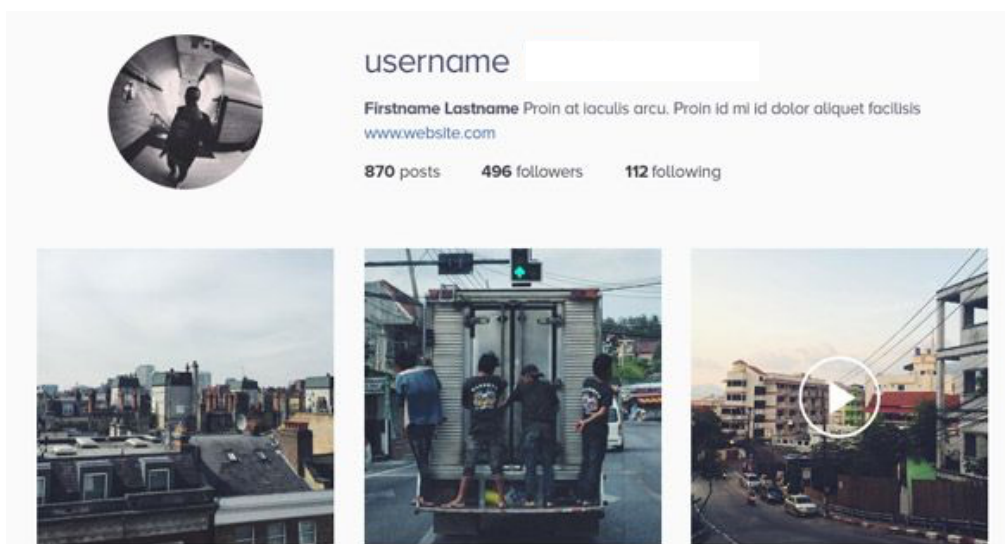
### Descripción

**Alpesgram** es una red social que permite a sus usuarios mostrar un conjunto de fotos en una estructura tipo matriz. Su tarea es generar el MVP de la aplicación enfocándose en la parte gráfica. En esta oportunidad usted debe hacer 3 pantallas: Home, Perfil y Ver Detalle Post. La aplicación se debe desarrollar en REACT.

### 1. Componentes (80%)

#### C1. Home (40%)

La primer pantalla que debe generar será la asociada al home de la aplicación. Esta página tal como se ve en la siguiente Figura cuenta con 2 secciones importantes: (1) la parte superior que contiene la información general del usuario y (2) el contenido central de la página.



<https://i.pinimg.com/originals/ab/3d/6d/ab3d6d329e4791eefd5bb20957d048c0.jpg>



# Programación con tecnologías web

## Parcial No. 1

Durante el proceso de desarrollo de la página usted debe tener en cuenta lo siguiente:

1. Respecto a la sección del usuario, usted debe crear el servicio mock que genere los datos requeridos del usuario: username, nombre completo, descripción corta, url, cantidad de posts, cantidad de seguidores, cantidad de cuentas seguidas.
2. Respecto a la sección central de la página, se deben mostrar siempre 12 imágenes, si la pantalla tiene un ancho mayor a 800 pixeles se debe mostrar una matriz de 3 filas y 4 columnas, si el ancho de la pantalla es mayor a 500px y menor a 800px se debe mostrar una matriz de 4 filas y 3 columnas, si la pantalla es más pequeña se debe utilizar una matriz de 6 filas y dos columnas. Todas la imágenes que se cargan se obtienen por medio de peticiones GET a la siguiente URL: <https://picsum.photos/350>

### C2. Perfil (25%)

Cuando el usuario haga click en la foto de perfil del usuario en el home, la página debe redirigir al usuario a la página del perfil. Esta vista debe lucir similar a la siguiente imagen:

The image shows a mockup of a user profile form. At the top, there is a circular profile picture placeholder. Below it, the form contains four text input fields, each with a label above it: 'Nombre de usuario' (with 'Username' as placeholder text), 'Nombre completo' (with 'Nombre completo' as placeholder text), 'Descripción del perfil' (with 'Descripción del perfil' as placeholder text), and 'URL página personal' (with 'URL página personal' as placeholder text). At the bottom of the form is a red-outlined button labeled 'Guardar Cambios'.

Los datos que se muestran en esta página deben ser extraídos del mismo servicio creado para el home. En esta pantalla debe aleatoriamente seleccionar si la información se presenta usando componentes de texto (<p>) o componentes editables (<input>).



# Programación con tecnologías web

## Parcial No. 1

### C3. Ver Detalle Post (15%)

La última vista que se debe realizar está enfocada en ver la información detallada de los post o imágenes que se ven en el Home. Por lo tanto, cuando el usuario haga click en una de las imágenes se debe mostrar un modal que muestre la foto en un tamaño más grande.

Dado que estamos usando datos generados aleatoriamente, para conseguir una imagen de un tamaño mayor puede usar la siguiente URL modificando el número al final que representa el tamaño en pixeles de la imagen solicitada: <https://picsum.photos/350>

### 2. I18N (10%) – Puede ser entregado fuera de clase (Domingo Media Noche)

Los *labels* estáticos de la aplicación deben ser internacionalizables. Asegúrese de soportar al menos dos idiomas (inglés y español).

### 3. ReadME (10%) – Puede ser entregado fuera de clase (Domingo Media Noche)

Agregue a su repositorio un archivo README que contenga un paso a paso para la ejecución de su parcial junto con un reporte de las decisiones y explicando el proceso de desarrollo de su parcial. Debe mencionar los componentes y elementos pertenecientes al ambiente de desarrollo de REACT que haya utilizado dentro de su solución.

### Entrega trabajo en clase

Cuando finalice el parcial suba los cambios a su repositorio de GitHub

Haga un [release](#) en su repositorio con la etiqueta v1.0.0 y el título **parcial\_1\_clase**.

Suba el enlace del *release* como respuesta a la actividad **Parcial1** de Bloque Neón.

Tenga en cuenta que no se revisaran cambios después de este *release*.

### OPCIONAL Entrega en casa

Si usted decide realizar en casa la entrega de los dos componentes finales de este examen en su casa, debe crear un nuevo release a más tardar el domingo a media noche. No es necesario realizar un nuevo envío.