# Laboratorio Final de Aplicaciones Ricas de Internet

UTEC Paysandú - 2023

## Objetivos

- Diseño de interfaces con HTML + CSS
- Practicar la manipulación del DOM
- Interactuar con una API pública predeterminada
- Usar transiciones, animaciones, transformaciones, variables y funciones CSS
- Practicar con alguna API moderna, definida a partir de ECMAScript (LocalStorage, fetch, canvas, etc.)

## Introducción

Bienvenidos a la startup "Bukowski", el emprendimiento que se preocupa por los escritores. En esta nueva empresa están buscando metodologías y aplicaciones novedosas que sirvan de soporte, inspiración y apoyo para los escritores. Usted llega aquí como prospecto a hacerse con el cargo de CTO de la empresa y quieren ponerlo a prueba a ver si sabe tanto como dice... Y quizás para robarle su trabajo, pero eso ya lo veremos a futuro.

La junta directiva del emprendimiento le pide que presente una app funcional que para nada está inspirado en la plataforma “750 words”: una aplicación web totalmente contenida en capa cliente que sirva de diario/blog y que inste a quién la use a escribir al menos 750 palabras al día en ella y sirva como herramienta para que el escritor le ponga ganas a su futura novela. Cada día a las 00 horas se abre una nueva página de este diario, guardando la anterior, para que el usuario pueda escribir las 750 palabras diarias de este día. La app le muestra al usuario cuánto tiempo dispone mediante una cuenta regresiva diaria que le indica cuánto falta para llegar a medianoche, además que debe mostrar en vivo un contador de palabras indicando la cantidad de palabras que se llevan escritas. Con cada cambio al post este debe ser persistido automáticamente.

El sistema debe ser multi usuario, manejando una lista de usuarios posibles tomados de una API REST, donde se podrá seleccionar alguno de estos usuarios y actuar como si fuese ese usuario, viendo los posts creados por ese usuario y pudiendo crear el post del día del usuario activo. Todo esto debe quedar persistido en LocalStorage (o en una API similar). Al volver a abrir la aplicación el último estado de la misma debe ser recuperado, lo que significa que debe ser el mismo usuario activo, se debe poder ver los mismos posts asignados a los usuarios y el último post editado debe estar igual que como se dejó.

El departamento de diseño de la startup le ha pasado los siguientes mockup de la app:

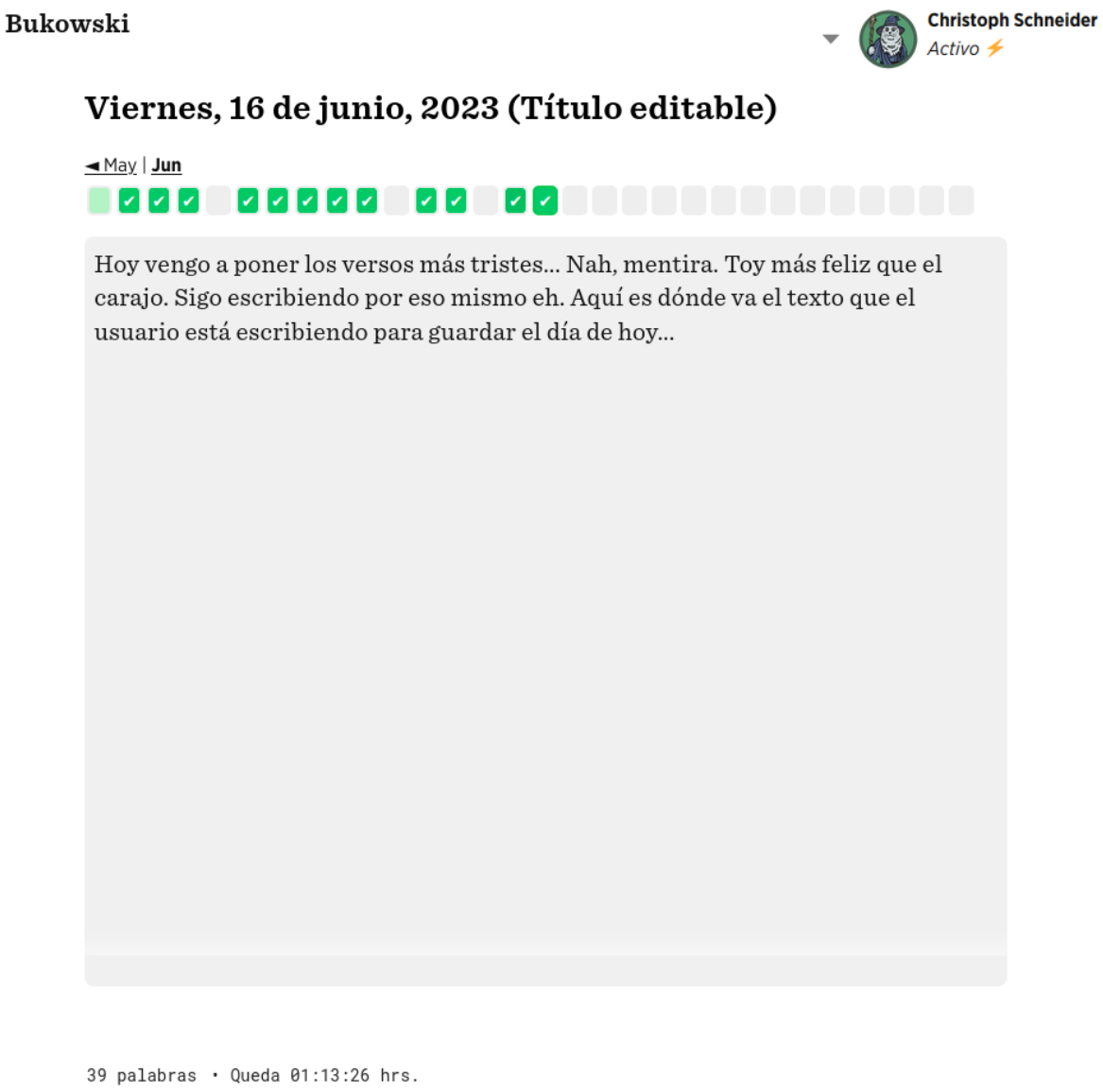


Figura 1.- Mockup de la webapp

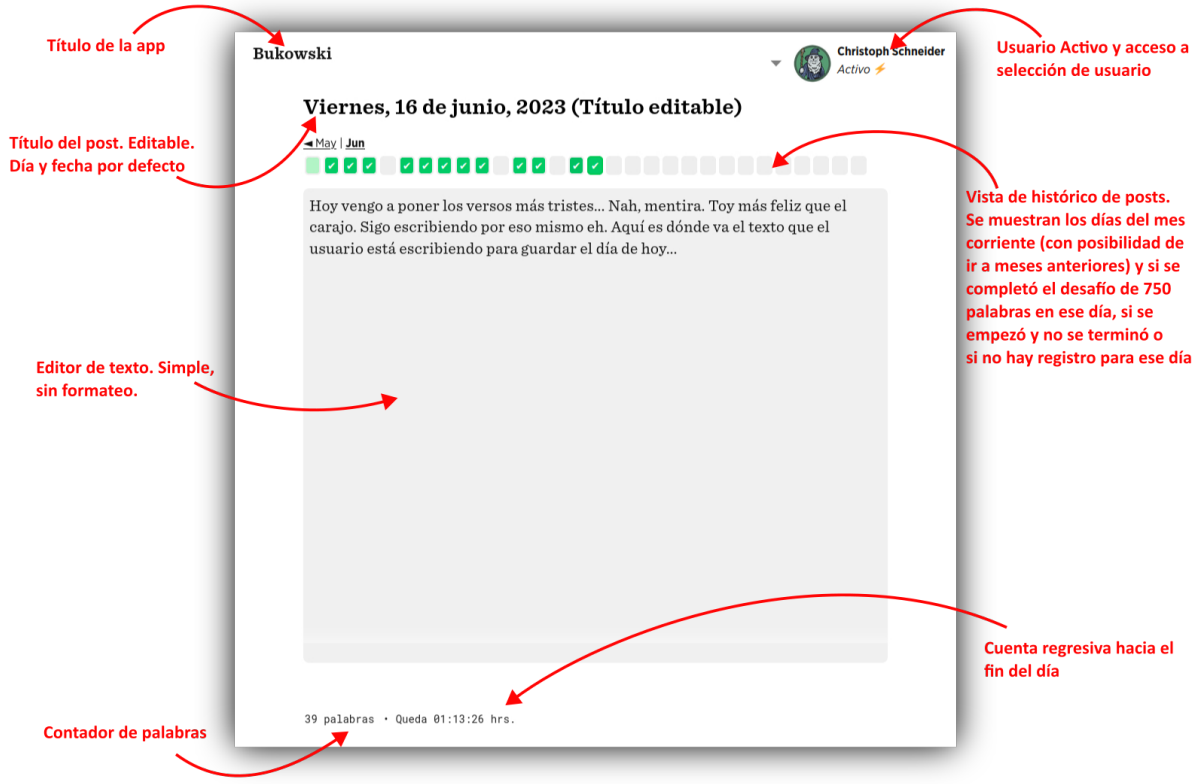


Figura 2.- Mockup con anotaciones

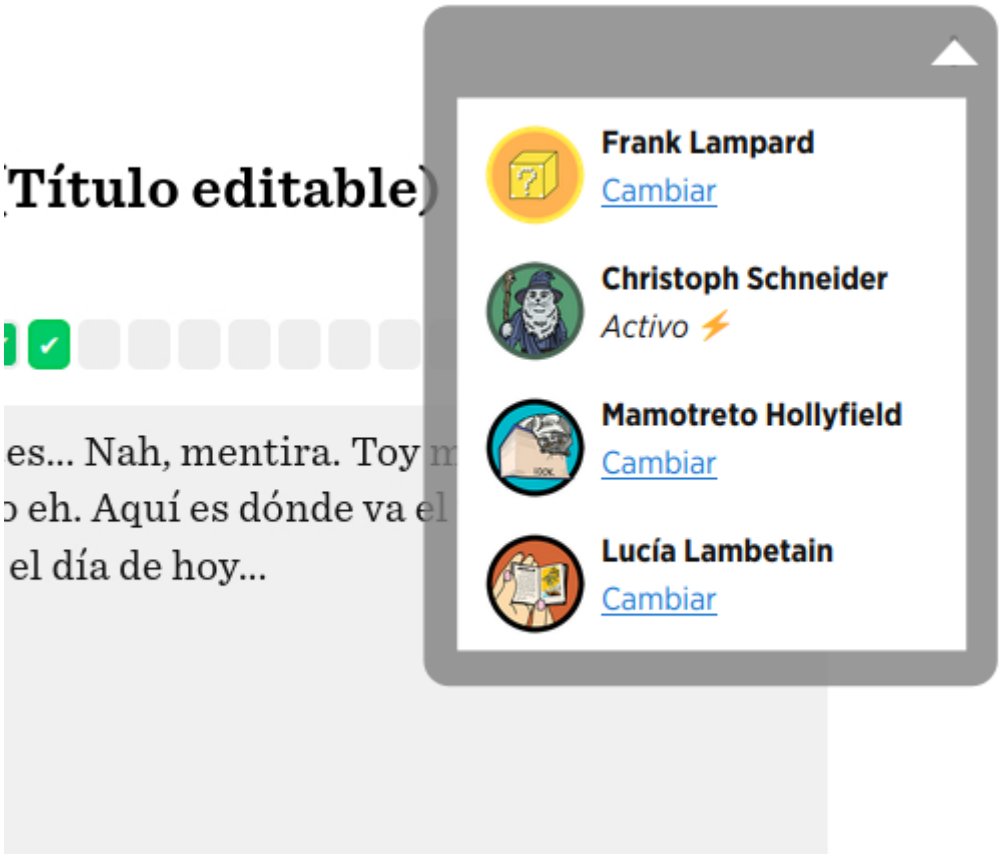


Figura 3.- Mockup del menú de selección de usuarios

# Nota del desarrollo

Todos los requerimientos a continuación, a excepción de los indicados como “OPCIONAL”, son requerimientos funcionales. A pesar de ello el que falte completar requerimientos no es causa de pérdida del laboratorio, tan solo se quitan puntos al resultado final.

# Requerimientos

- Generar un layout en HTML/CSS que respete el presentado por el departamento de diseño. No tiene porqué ser exactamente ese diseño, pero sí que respete la ubicación de lo distintos ítems que la componen y que aparezcan todos los componentes pedidos.
- La “lista de usuarios” debe cargar una lista de usuarios utilizando la API REST JSON Placeholder (http://jsonplaceholder.typicode.com/) y mostrar nombre de usuario. Al hacer click en uno de estos usuarios quedará habilitado como “usuario activo”, dejándolo como usuario activo para cualquiera de las demás acciones que se pueden realizar.
- La forma de presentar el componente de selección de posts anteriores y cómo se muestran los estado de estos días queda a decisión del desarrollador. Lo que si es importante es tener en cuenta que un día puede tener tres estados: Sin post, Incompleto (con un post, pero con menos de 750 palabras) y Completo (con un post con 750 palabras o más).
- La webapp debe mantener su estado entre sesión y sesión, por lo que hay que generar una capa de persistencia en la webapp, utilizando la API LocalStorage o IndexedDB.
- Si los posts anteriores pueden ser editables o no queda a criterio del desarrollador. La idea de la app es que no sean editables, pero no implica un error considerar la edición de posts.
- OPCIONAL: agregar animaciones a las distintas acciones o cambios de estado de la aplicación (se llega al objetivo del día, se abre el menú, etc.).
- OPCIONAL: crear una aplicación instalable utilizando el estándar de progressive web apps (PWA).

# Restricciones

- La aplicación web a desarrollar no debe ni puede depender de librería externa alguna
    - No obstante se permite el uso de la librería de iconos Font Awesome o similar, siendo la aplicación de estos íconos algo no excluyente ni necesario.
    - También se pueden utilizar fuentes cargadas con @font-face, utilizando algún servicio externo como Google Fonts o con fuentes descargadas.
- La aplicación web debe ser lo más autocontenida posible, tratando de que la cantidad de archivos necesarios sean mínimos y que haya una estructura de directorios correcta. Se recomienda que se entreguen:

    - Un archivo 'script.js', conteniendo la lógica necesaria para que la aplicación web se inicialice y funcione, escrito preferiblemente en ECMAScript. En caso de ser escrito en un lenguaje pre-compilado (Dart, Coffeescript, etc.) incluir los archivos originales y recordar: **no se deben usar librerías Javascript externas**. El archivo generado debe ser puro, sin dependencias en librerías.
    - Un archivo 'styles.css', conteniendo los estilos necesarios para la correcta visualización y funcionamiento de la aplicación web. En este caso no se permite usar lenguajes pre-compilados (Less, Sass, etc.)
    - Un archivo 'index.html', siendo el archivo que contiene el marcado HTML de la aplicación web.
    - En caso de generar una PWA todos los archivos de manifiesto necesarios.
- Cualquier otro archivo extra no restará puntos. Se pueden incluir imágenes, archivos de configuración o lo que sea necesario. Tan solo **se recomienda** minimizar la dependencia en muchos archivos.
- Y por último y principal: El laboratorio es **individual**.

# Fecha de entrega

La fecha límite de entrega de los archivos del laboratorio será el **domingo 16 de Julio, a las 23:59 hrs. + 1 minuto**, a través del formulario de entrega del laboratorio en la plataforma EV1 de UTEC. Es preferible que se entregue un único archivador comprimido. En caso de problemas en la entrega se puede enviar el trabajo vía e-mail a la dirección maximiliano.fernande@utec.edu.uy (Prestar atención que el apellido en la dirección de e-mail es "fernande", sin Z)