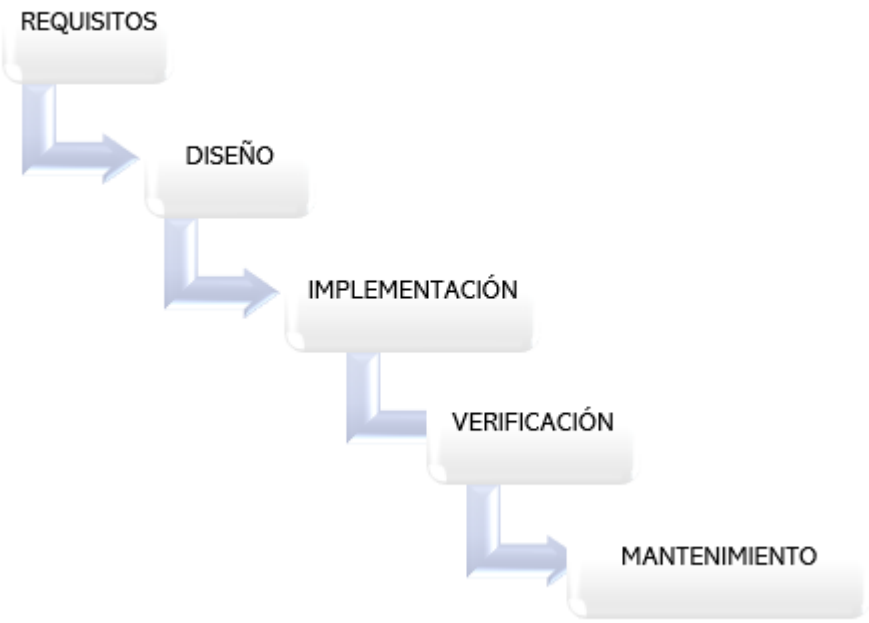
Primeros Pasos React: ***El tablero Kanban***

Fechas de entrega: 15 de Noviembre de 2021

Contexto

Cuando nos encontramos en un proyecto, una de las partes importantes de ese proyecto es la forma en la que se gestiona, lo que conocemos como metodología

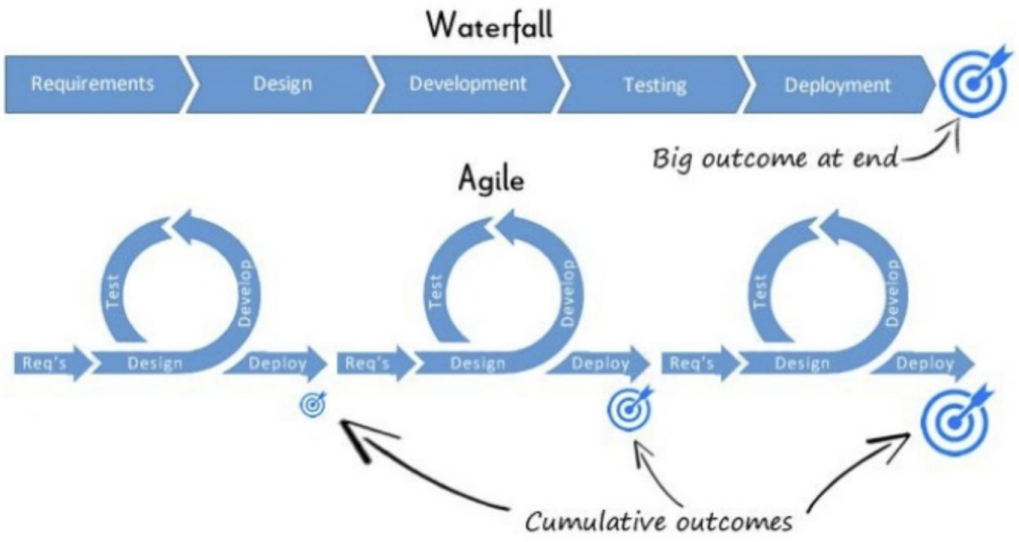
Antiguamente se gestionaban los proyectos de software utilizando una metodología ***waterfall***, que consistía en dividir TODO el proyecto en 5 fases y hasta que no estuviese acabada esta fase no podía empezar con la siguiente. Las fase solían ser:



El problema de esta metodología es que si había algún cambio durante la ejecución del proyecto, este tendría que esperar a que acabaran las fases para poder aplicar el nuevo cambio en nuestro proyecto de software.

El 12 de febrero de 2001, diecisiete críticos de los modelos del desarrollo de software se juntaron para crear lo que se conoce como ***Manifiesto Agile***, un documento que fue el germen de las metodologías que se usan para desarrollar proyectos, las conocidas metodologías ágiles.

Estas metodologías dividen el proyecto en sprints (periodos de tiempo donde quiero desarrollar una pequeña funcionalidad que aporta valor a un usuario)
Esto permite que de un sprint a otros sprint pueda cambiar las prioridades de mi negocio y por tanto las de mi desarrollo.



Existen muchas metodologías de desarrollo agile, pero las dos más conocidas son:

- SCRUM
- KANBAN

En ambas metodologías tienen un sitio para colocar sus tareas y en el estado en el que están las tareas.

Para Kanban, principalmente las tareas tienen 3 estados: TODO, INPROGRESS y DONE.

Objetivos

Desarrollar una aplicación web para que cualquier equipo del mundo pueda gestionar su proyecto con una metodología KANBAN

Requisitos

Para realizar esta práctica se han de tener conocimientos de:

- HTML
- CSS
- JS
- ReactJS:
    - Creación de una aplicación
    - Creación de componentes
    - Consumir un componente dentro de otro componente
    - Pasar propiedades de un componente padre a un componente hijo
    - Utilizar el estado de un componente de reactjs
    - Pasar información de un componente hijo a un padre
    - Ciclo de vida de un componente ReactJS

Diseños

La aplicación se debe visualizar como las siguientes imágenes:

En la imagen podemos ver que hay 3 columnas que representan el estado en el que está la tarea. Las columnas tienen un botón más para añadir tareas a esa columna

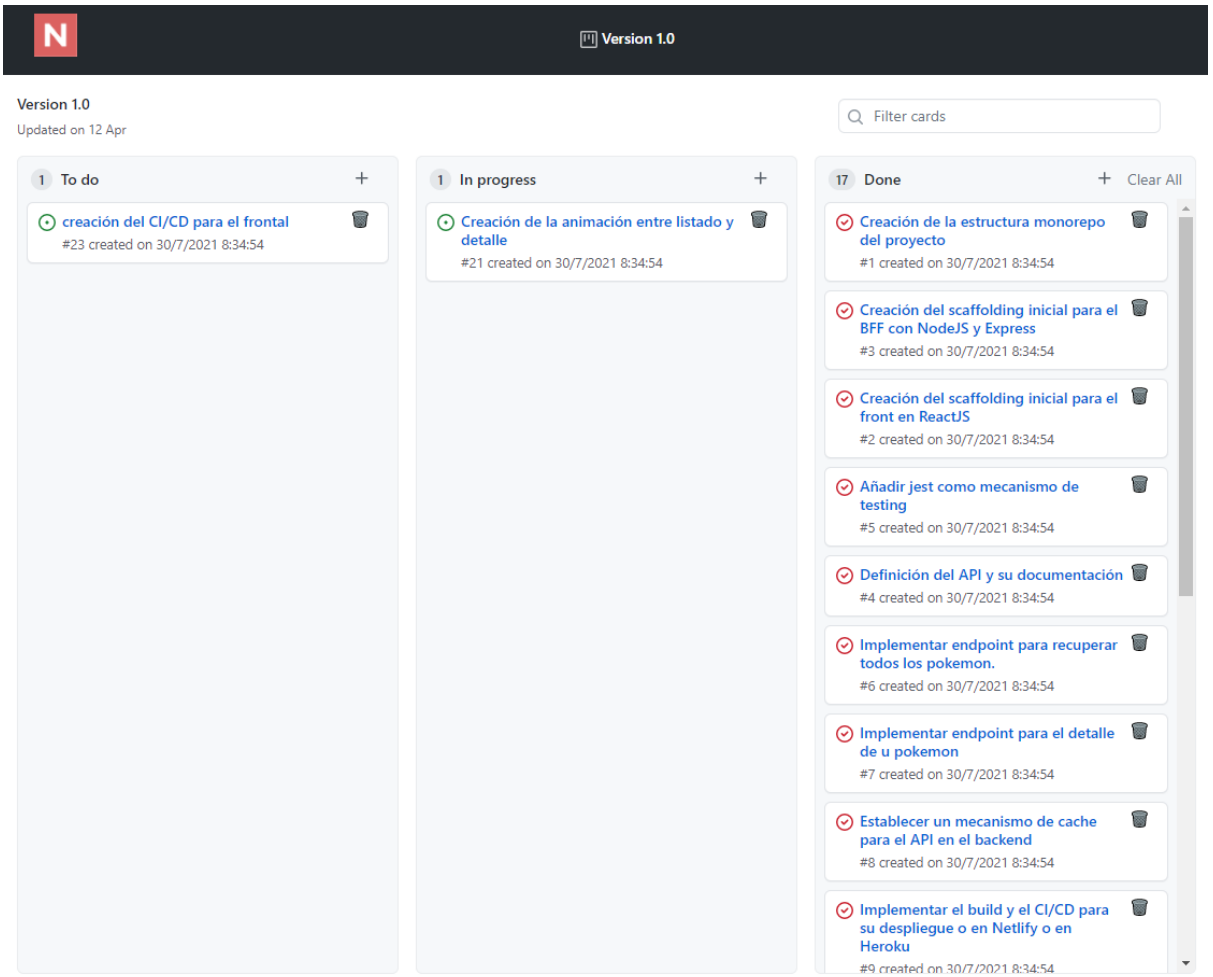
Además las columnas muestran el número de tarjetas que tienen y si hay overflow, se genera un scroll interno. LA PAGINA NUNCA TENDRÍA SCROLL

Cada tarea se muestra con una card con su título, su id y la fecha en la que se ha creado
Las tareas tienen un botón de eliminar, que eliminará definitivamente esa tarea.

Las tareas en TODO e INPROGRESS tienen un icono de pending (icono en verde)

Las tareas en DONE se encuentran cerradas y por eso tienen un icono en rojo

Además la página tiene un buscador que cuando escribamos 3 caracteres o más, nos mostrará solo las tareas de cada columna que contienen en su título esa búsqueda

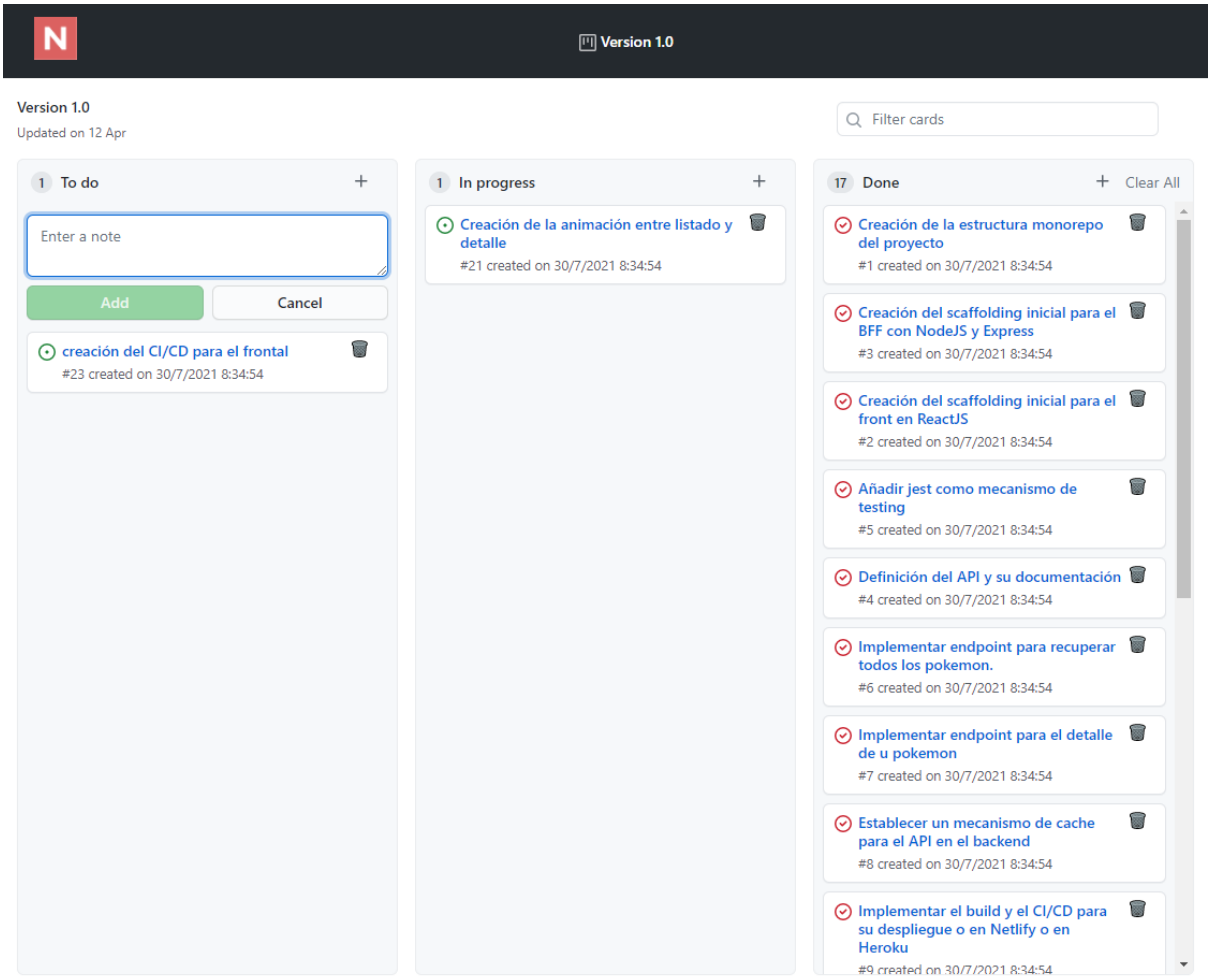


Cuando se pulsa al botón añadir de una columna, nos saldrá un formularió para poder añadir la tarea.

El botón aceptar aparece deshabilitado mientras el input para poner el título de la tarea esté vacío.

Si se le da al botón aceptar, el formulario desaparece y aparece la nueva tarea creada

Si se le da a cancelar, el formulario desaparece



Fuente

La página tiene la siguiente fuente:

`-apple-system,BlinkMacSystemFont,Segoe UI,Helvetica,Arial,sans-serif,Apple Color Emoji,Segoe UI Emoji;`

Colores

- Color de fondo del header: `#24292e`
- Color de fondo cada columna de tareas: `#f6f8fa`
- Color de los títulos de las tareas: `#0366d6`
- Color del icono de tareas en progreso: `#22863a`
- Color del icono de tareas terminadas: `#cb2431`
- Color de fondo del botón añadir: `#2ea44f`
- Botón añadir deshabilitado: `mismo color pero con opacidad 0.7`
- Color de fondo del botón cancelar: `#fafbfc`
- Color de borde de las tarjetas y el botón cancelar: `rgba(27,31,35,0.15)`

Imágenes

- Neoland logo: https://www.neoland.es/hubfs/favicon%20neoland-02-02-02.png
- Icono papelera: `podéis usar un emoji`
- Icono tarea pending: `Icono de material design llamado `pending``
- Icono de tarea cerrada: `Icono de material design llamado `check circle``

Funcionalidades

Aplicación Base

La aplicación base tiene que tener la siguientes funcionalidades:

- Al iniciar la aplicación se mostrará la cabecera, el título y el buscador y la última actualización del tablero que será la fecha y hora actual
    - Esta fecha irá cambiando cada vez que se haga una modificación (crear una card)
- Se crearán las 3 columnas con cada uno de los estados con 0 tareas en cada uno. Las columnas tienen que ocupar el espacio restante de la pantalla (tienen que tener un height) y no pueden generar scroll en la página.
- Poder crear tareas en cada una de las columnas con la siguiente información
    - Título
    - Estado (pending o done)
    - id
    - Fecha de creación
- Cuando creemos una tarea, se actualizará el número de tareas de cada columna.
- Si la tarea se encuentra en DONE su estado será done, sino será pending y se deberá mostrar con su correspondiente icono
- Se debe permitir borrar cada una de las tareas de manera independiente al hacer click en su icono de borrar
- En la columna de DONE, al hacer click en “clear all”, se eliminarán todas las tareas que contenga y la columna quedará vacía
- Al buscar en el input del buscador, se debe mostrar en cada columna las tareas que contengan en su título lo que el usuario introdujo en el input de búsqueda

Guardando el estado de la aplicación (Medium)

- Cuando se refresque la aplicación o cuando se cierre el navegador, la web me deberá mostrar el estado en el que dejé mi tablero Kanban

Cambiar de estado las tareas (Leyenda) (Opcional)

- Se debe permitir que el usuario cambie una tarjeta de columna arrastrando la tarea de su columna original y soltándose en la columna destino.

- Este tipo de funcionalidad es conocida como Drag&Drop.
- Para poder implementar esta funcionalidad existen dos opciones:
    - Utilizar los eventos nativos de Drag&Drop de JavaScript en nuestra aplicación ReactJS.
        - Tutorial en VanillaJS
        - Tutorial js nativo con ReactJS
    - Utilizar una librería que nos pueda ayudar con toda la gestión nativa.
        - react-draggable - npm (npmjs.com)
        - Tutorial

Entrega

La entrega se realizará el 15 de Noviembre y se expondrá a los compañeros en clase.

Se debe entregar la URL pública de la aplicación reactJS generada con Github Pages.

Os dejo un tutorial con una manera diferente a como hemos enseñado en clase por si os sirve. How to deploy React App to GitHub Pages - DEV Community