



BANCO PICHINCHA

En Banco Pichincha nos estamos transformando. Nuestros 114 años de trayectoria, nos ratifican que contar con el mejor talento humano es imprescindible para lograr nuestros objetivos.

Lo que une e identifica a nuestros 5.400 colaboradores son los valores que marcan nuestro día a día: integridad, responsabilidad, servicio al cliente, productividad e innovación y solidaridad. Por ello te invitamos a ser parte de nuestra evaluación para calcular el nivel de seniority de nuestros candidatos en el área de desarrollo frontend.

Enunciado

El siguiente ejercicio tiene como finalidad el desarrollo de un gestor de tareas (un todo list), el cual permita al usuario del mismo listar, crear, actualizar y eliminar sus tareas, así como buscar tareas según determinados filtros.

Instrucciones

- **Manejo de Git**
 - Debe crear un repositorio público en el cual se comparta el código implementado para la resolución del problema
 - Al finalizar el ejercicio y una vez que los cambios se encuentren en el repositorio público se deberá crear un Pull Request de los cambios hacia la rama del repositorio compartido asignada con su nombre de usuario
- **Diseño**
 - El desarrollador deberá maquetar la aplicación según el diseño y condiciones presentadas
 - La implementación de estilos debe realizarse sin uso de librerías CSS de terceros como bootstrap, tailwindcss, etc.
 - Es prioritario que se implemente el diseño con la filosofía mobile first, por lo que el diseño mobile es prioritario, el diseño desktop es solo referencial, no se toma en cuenta la adaptación de diseño en diferentes dispositivos

Nota: Es posible realizar la implementación de estilos con librerías como styled-components y similares, en donde el programador continúa escribiendo código CSS para establecer sus estilos.

- **Codificación**

- Para llevar a cabo la implementación del ejercicio en esta ocasión se contará con un proyecto base, el cual cuenta dependencias instaladas, componentes ya desarrollados y con unas cuantas funcionalidades, la idea es que en base al proyecto base proporcionado, el programador complete el ejercicio tomando en cuenta lo siguiente.
 - Debe hacer uso obligatorio de los componentes y funcionalidades, proporcionadas por el proyecto base y con ayuda, modificación o expansión de las mismas completar el ejercicio.
 - El proyecto base cuenta con una suite de dependencia instaladas el programador puede hacer uso de ellas o en su defecto puede agregar cuales crea necesarias no hay ninguna restricción sobre esto.

- **Pruebas unitarias**

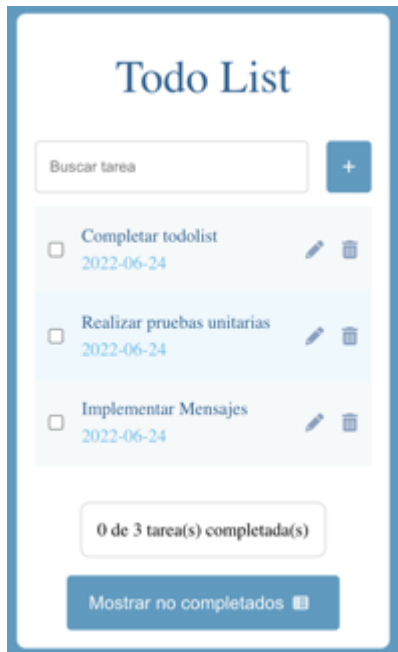
- La implementación de pruebas unitarias es sumamente importante, es requerido que la implementación de las funcionalidades tenga sus respectivas pruebas.
- El proyecto base trae consigo un directorio destinado para los test unitarios, en él se aloja un único archivo, el cual contiene los test que deben ser implementados por el programador.

Nota importante

Como se menciona en la sección instrucciones, las pruebas unitarias son sumamente importantes por lo que tendrán una ponderación mayor para los criterios de evaluación, además cada funcionalidad o criterio de aceptación que no cuente con su prueba respectiva no será tomado en cuenta para la calificación.

Criterios de aceptación

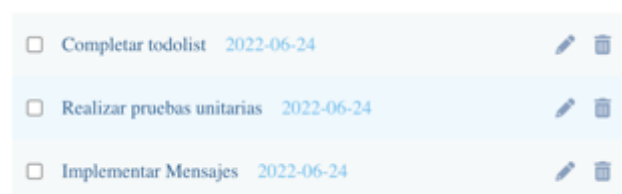
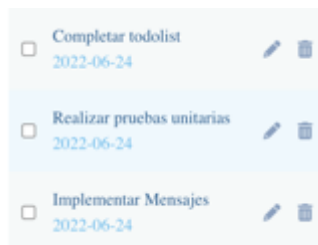
- Implementar el listado de tareas según el siguiente diseño, el cual es mobile first.



Mobile

Desktop

Cada tarea debe mostrarse de la siguiente manera, checkbox para control del cumplimiento de la tarea, descripción de la tarea, fecha límite de la tarea, un botón para editar y uno para eliminar la tarea.



Mobile

Desktop

- Implementar la búsqueda de tareas por descripción (búsqueda solo en memoria)

- Implementar una barra de progreso de las tareas completadas, la cual se va pintando con respecto al porcentaje de cumplimiento de las tareas.

0 de 3 tarea(s) completada(s)







1 de 3 tarea(s) completada(s)

- Implementar un filtro para mostrar las tareas faltantes por completar, a partir del siguiente botón, el cual al dar clic por primera realiza el filtrado de las tareas no completadas y su texto cambia a “Mostrar todos”. Si es texto del botón es el de “Mostrar todos” cuando se da clic sobre el botón entonces se quita el filtro y se deben desplegar todas las tareas registradas (Filtro solo en memoria).

Mostrar no completados

Mostrar todos

- Implementar el control de estado de completitud de las tareas, esto se debe realizar a través del checkbox que forma parte de una tarea (un todo), Si una tarea es marcada como completa esta debe cambiar su estado, el cual es manejado por el atributo status. Además, si la tarea se encuentra completada entonces esta debe ser tachada para una mejor visualización.

<input checked="" type="checkbox"/>	Completar todolist 2022-06-24	 
<input type="checkbox"/>	Realizar pruebas unitarias 2022-06-24	 
<input type="checkbox"/>	Implementar Mensajes 2022-06-24	 

- Implementar la creación y actualización de tareas a partir de un formulario según el siguiente diseño. El cual requiere de una descripción y una fecha límite para completar la tarea.



Formulario 'Todo List' para crear una tarea. El formulario tiene un título 'Todo List' en azul. Debajo del título, hay un campo de texto con el placeholder 'Descripción'. Debajo de este, hay un campo de fecha con el placeholder 'dd/mm/aaaa' y un icono de calendario. En la parte inferior, hay dos botones: 'Volver' (gris) y 'Agregar' (azul).

- La descripción y la fecha límite son de orden obligatorios por lo que es requerido que se muestren mensajes que indiquen que los campos son requeridos.



Formulario 'Todo List' con mensajes de error. El formulario tiene un título 'Todo List' en azul. Debajo del título, hay un campo de texto con el placeholder 'Descripción'. Debajo de este, hay un mensaje de error en rojo: 'Descripción es requerida'. Debajo de este, hay un campo de fecha con el placeholder 'dd/mm/aaaa' y un icono de calendario. Debajo de este, hay un mensaje de error en rojo: 'Fecha límite es requerida'. En la parte inferior, hay dos botones: 'Volver' (gris) y 'Agregar' (azul).

- Implementar el registro de una nueva tarea, esto debe realizarse a partir del botón '+', el cual se señala a continuación. al dar clic sobre el botón '+', la aplicación debe redireccionar hacia el formulario mencionado anteriormente.



- Implementar la edición de una tarea, esto debe realizarse a partir del botón editar, el cual se señala a continuación. al dar clic sobre el botón editar, la aplicación debe redireccionar hacia el formulario mencionado anteriormente, con los datos respectivos de la tarea seleccionada.

☐ Nueva tarea 2022-06-24



- Implementar la eliminación de una tarea, esto debe realizarse a partir del botón eliminar, el cual se señala a continuación, al dar clic se debe realizar la eliminación de la tarea seleccionada.

☐ Nueva tarea 2022-06-24



- Implementar el control que cuando la lista de tareas se encuentre vacía debe mostrarse el siguiente mensaje, según diseño.

Todo List





No tienes tareas registradas

API Docs

La documentación de los servicios necesarios para realizar el ejercicio es el siguiente <https://documenter.getpostman.com/view/21329689/Uz5JHFLY>

