

Desarrollo de Sistemas Web (Front End) - 2° A

2do Cuatrimestre 2024

Trabajo Práctico Grupal 1: Tarjetas Personales en React

Consigna:

En este trabajo grupal, un integrante del equipo será responsable de realizar un fork del proyecto proporcionado por el profesor en Replit. A partir de este fork, los demás integrantes deberán agregar sus propios componentes React, que representarán tarjetas personales con información de su perfil. El objetivo es trabajar en equipo, implementando componentes reutilizables y manteniendo una estructura de código clara y modular.

- Fecha lanzamiento de trabajo práctico: 14/10/24
 - Fecha de entrega: 28/10/24
-

Detalles del Trabajo:

Creación del Fork del Proyecto:

1. Un integrante del grupo deberá asumir el rol de administrador y realizar el fork del proyecto React en Replit que se encuentra en el siguiente link:

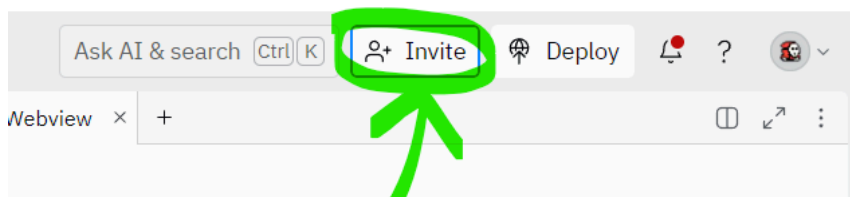
<https://replit.com/@lucianomartin11/BASE-TP1?v=1>

2. Los demás integrantes deberán sumar sus componentes al proyecto trabajando en colaboración con el equipo.
 3. A partir de ese fork, trabajarán integrando los componentes de cada miembro del equipo en una aplicación principal.
-

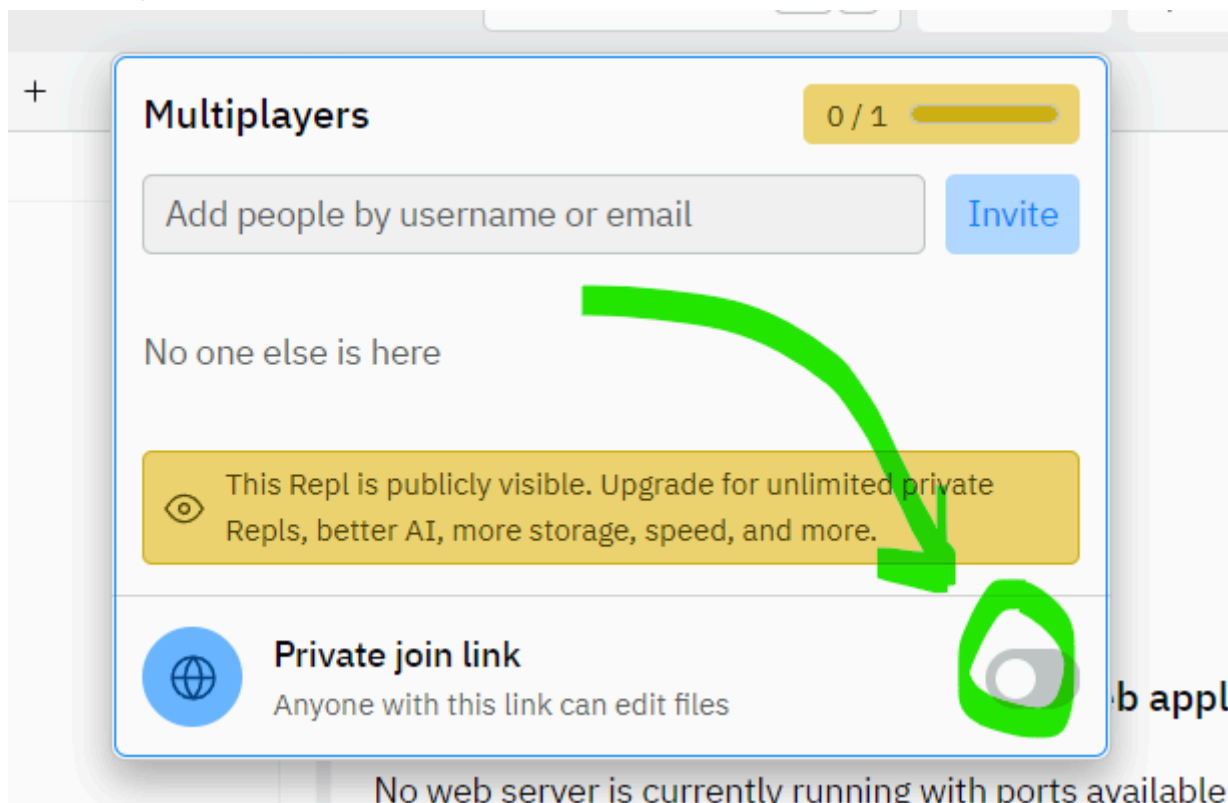
Como entregar el trabajo:

El integrante administrador del proyecto deberá hacer un "Private link" y compartirlo con el docente.

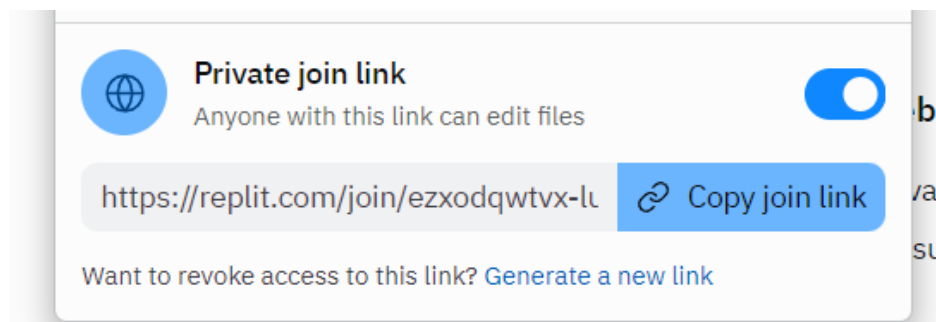
- 1 - Hacer clic en INVITE en la parte superior derecha



2 - Activar la opción "PRIVATE JOIN LINK"

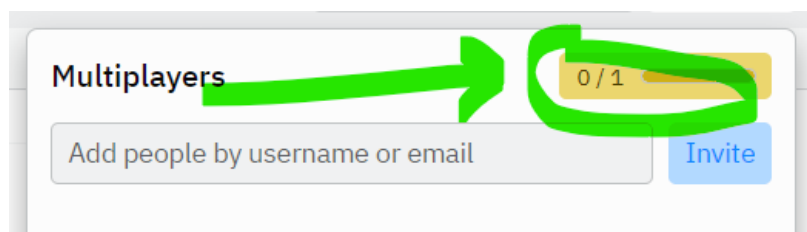


3 - Copiar y pegar de forma privada con el docente el link generado:



IMPORTANTE:

Replit solo permite añadir 1 colaborador a cada proyecto de forma gratuita



Por lo cual solo el estudiante administrador y el docente tendrán acceso al proyecto.

Tenga cuidado que solo al hacer click te añade como administrador, por este motivo el link debe ser compartido de forma privada con el docente.

REQUERIMIENTOS:

Componentes:

Cada integrante debe crear un componente React que funcione como una tarjeta personal. Esta tarjeta debe cumplir con los siguientes requisitos mínimos:

Información a mostrar:

- Nombre completo.
- Imagen de perfil (puede ser una URL de imagen o un avatar genérico).
- Descripción breve (2-3 líneas sobre ti o algún dato interesante).
- Enlace a un perfil social o algún enlace relevante (por ejemplo, LinkedIn, GitHub, etc.).

Estilos:

- Utilizar CSS para dar un diseño atractivo a la tarjeta (puede ser usando clases o styled-components).
- La tarjeta debe tener un diseño responsive y un formato adecuado para dispositivos móviles.

Modularidad:

- El componente debe ser modular y fácil de integrar en la aplicación principal.

Integración en la Aplicación Principal:

- El integrante que realizó el fork del proyecto será responsable de crear el componente principal que renderizará las tarjetas de todos los integrantes del equipo.
- Este componente principal deberá importar y utilizar los componentes de tarjetas creados por cada miembro del grupo.
- No es necesario que todos los componentes sumados por los integrantes del equipo tengan el mismo estilo, pueden ser estilos distintos lo cual refuerza el concepto de modularidad.

Funcionalidades adicionales (opcional):

Si el grupo lo desea, pueden agregar funcionalidades adicionales como:

- Uso de rutas: Implementar el uso de rutas con **react-router-dom** para alternar entre las tarjetas de cada integrante, de modo que al hacer clic en un nombre o tarjeta, se pueda navegar hacia una vista individual de cada tarjeta.
 - Configurar el componente de la tarjeta para recibir datos como **props** (nombre, descripción, enlace, imagen) si es necesario, asegurando la transmisión dinámica de la información desde el componente principal.
 - Incluir una interacción con el usuario, como cambiar el color de fondo al pasar el mouse o mostrar más información al hacer clic, si mejora la experiencia del componente.
-

Entrega:

- Fecha de entrega: 28/10/24
 - Presentación del trabajo:
 - Cada grupo deberá enviar el enlace al fork en Replit con el proyecto completamente funcional.
-

Aclaración para estudiantes:

Es importante tener en cuenta que un proyecto realizado y probado en VS Code puede no funcionar en Replit. Se recomienda trabajar desde cero en Replit para asegurar la compatibilidad y el correcto funcionamiento del proyecto. (**Ver apéndice 1**)

Evaluación:

- Funcionalidad de los componentes: Cada tarjeta debe cumplir con los requisitos solicitados.
 - Diseño: Las tarjetas deben ser atractivas y responsive.
 - Colaboración grupal: Se evaluará la integración de los componentes y la organización del código.
-

Apéndice 1 :

Diferencias entre el desarrollo de React en VS Code y Replit

1. Entorno de desarrollo:

- **VS Code:** Entorno local, requiere instalación de Node.js y npm.
- **Replit:** Entorno en línea preconfigurado, sin necesidad de instalación local.

2. Configuración de dependencias:

- **VS Code:** Control total sobre la instalación y versiones de dependencias.
- **Replit:** Puede haber limitaciones en la instalación de ciertas dependencias o incompatibilidades.

3. Transpilación de JSX:

- **VS Code:** Soporta JSX directamente con herramientas como Babel o `create-react-app`.
- **Replit:** A veces requiere configuraciones específicas para manejar JSX correctamente.

4. Ejecución de la aplicación:

- **VS Code:** Se ejecuta en un servidor local, necesita comandos de terminal para iniciar.
- **Replit:** Se ejecuta en el navegador directamente, facilitando el acceso y la colaboración.

5. Almacenamiento y control de versiones:

- **VS Code:** Requiere sistemas de control de versiones como Git para colaboración.
- **Replit:** Incluye funciones de colaboración en tiempo real, sin necesidad de configuración adicional.

6. Compatibilidad del proyecto:

- Un proyecto desarrollado y probado en **VS Code** puede no funcionar en **Replit**, por lo que se recomienda empezar desde cero en Replit para asegurar la compatibilidad.