

Prueba técnica para postulantes (Fullstack)

La siguiente es una prueba para evaluar a los postulantes enfocada en desarrollo backend y frontend.

Introducción

Este repositorio contiene una serie de requerimientos, que buscan evaluar las capacidades técnicas del candidato con respecto a las principales funciones y responsabilidades que se requieren dentro del área de desarrollo de grupo Coppel.

¿Qué se busca evaluar?

Principalmente los siguientes aspectos:

- Estructuración de una aplicación: Conocimiento e implementación de patrones de arquitectura y de diseño tanto en frontend como en backend.
- Calidad del código entregado: Conocimiento y uso de buenas prácticas de codificación.
- Familiaridad con los entornos de desarrollo: Conocimiento y uso de las herramientas de desarrollo backend y frontend más actuales y de mayor uso.

Instrucciones previas al desarrollo

1. Se requiere de una cuenta de GitHub para realizar esta prueba.
2. La prueba debe ser realizada preferentemente con python o javascript.
3. Antes de comenzar a programar:
 1. Realice un Fork de este repositorio.
 2. Clone el fork a su máquina local.
 3. Cree una rama con su nombre completo en su repositorio local.
 - ej: git checkout -b ERNESTO_HERNANDEZ_CHAVEZ
4. Al finalizar, existen 2 (dos) opciones para entregar su proyecto:
 1. [Pull Request](#):
 1. Hacer un push de su rama local a github.
 2. Crear un Pull Request de la rama con su NOMBRE al repositorio de origen.
 3. Notificar de la creación del pull request al correo de contacto que se le haya proporcionado.
 2. Entrega por correo:
 1. Crear un archivo comprimido (.zip o .rar) de su proyecto.
 2. Enviar el zip como archivo adjunto al correo de contacto que se le haya proporcionado.
5. El proyecto deberá ser expuesto mediante una PaaS (heroku, gcloud, aws, ...) y los links correspondientes deberán ser incluidos en la descripción del pull request o en el email donde se entregue el proyecto.
6. En caso de ser necesario se debe incluir instrucciones, llaves de apis de terceros o cualquier otro dato necesario para correr la aplicación en un archivo README.md.

Aplicación TODO

Se requiere generar el backend y el frontend necesario para una aplicación de registro y control de tareas que proporcione las siguientes funcionalidades principales:

- Registro del usuario en la aplicación.
- Manejo de sesión del usuario (Abrir y cerrar sesión).
- Manejo y visualización de tareas (Crear, editar y borrar).

Pantallas

A continuación se presenta el listado de las pantallas y los elementos mínimos que deben contener.

Pantalla de inicio de sesión.

- Input de nombre de usuario.
- Input de contraseña.
- Botón de inicio de sesión.
- Botón de Login.

Logo

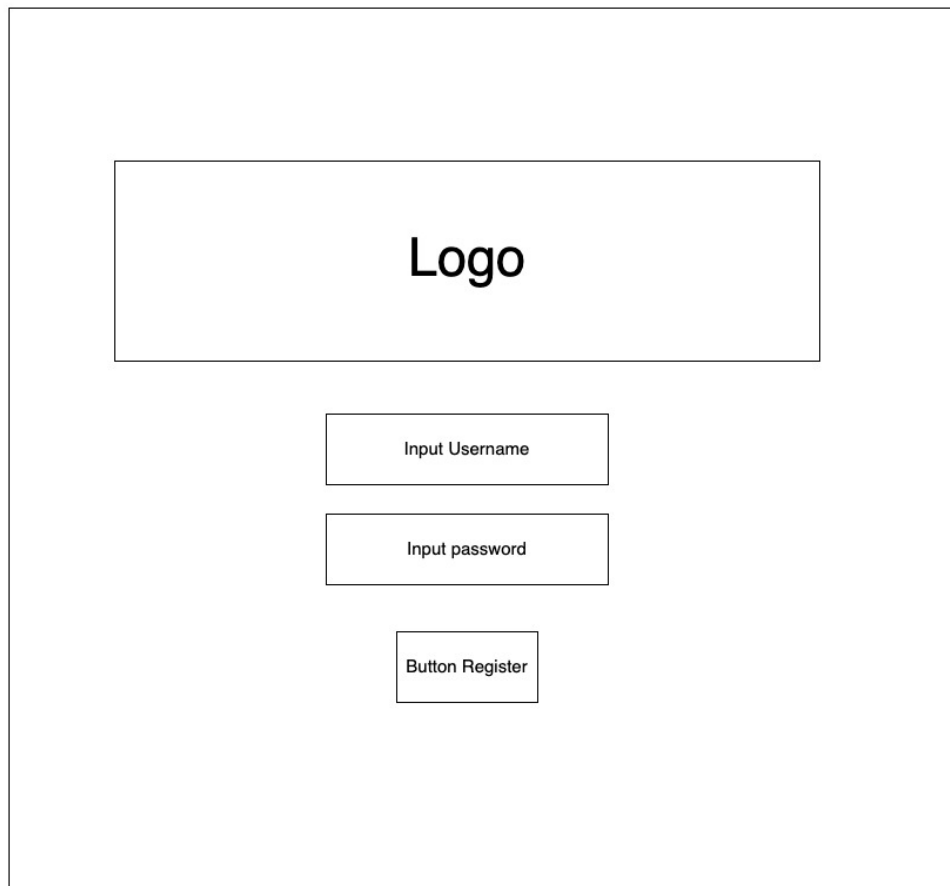
Input Username

Input password

Button Login

Pantalla de registro:

- Input de nombre de usuario.
- Input de contraseña.
- Botón de registro.



The image shows a registration form layout within a large rectangular container. At the top center is a large rectangular box labeled "Logo". Below the logo, centered, are three input fields stacked vertically: "Input Username", "Input password", and "Button Register".

Pantalla principal:

- Menú:
 - Botón de cierre de sesión.
- Listado de elementos.
 - Elemento:
 - Contenido.
 - Botón de edición.
 - Botón de borrado.
- Botón de creación de nueva tarea.

Button Logout

Task 1

Button Edit

Button Delete

Task 2

Button Edit

Button Delete

Input New task

Button Add Task

Casos de uso:

Registro

1. El usuario accede a la pantalla de login y elige la opción de registro.
2. En la pantalla de registro el usuario introduce correo válido y una contraseña alfanumérica.
3. La aplicación realiza el proceso de registro.
4. La aplicación redirige al usuario a la pantalla de inicio de sesión.

Inicio de sesión

1. El usuario accede a la pantalla de login e introduce los datos con los que se registró (correo y contraseña).
2. La aplicación valida los datos del usuario.
3. La aplicación redirige al usuario a la pantalla principal.

Cierre de sesión

1. El usuario con sesión válida en la pantalla principal da click al botón de cierre de sesión.
2. La aplicación limpia la sesión del usuario.
3. La aplicación redirige al usuario a la pantalla de inicio de sesión.

visualizar tareas

1. El usuario con una sesión válida accede a la pantalla del listado de tareas.

2. La aplicación muestra la lista de tareas del usuario.

Creación de tareas

1. El usuario dentro de la pantalla principal selecciona la opción de crear nueva tarea.
2. La aplicación mostrará un modal con un formulario para la creación de la tarea.
3. El usuario introduce el contenido de la tarea y da click en el botón crear.
4. La aplicación registra la tarea en el backend y la agrega a la lista en la pantalla.

Edición de tareas

1. El usuario selecciona la opción de editar de la tarea.
2. La aplicación mostrará un modal con un formulario para la edición de la tarea.
3. El usuario modifica el contenido y da click en el botón actualizar.
4. La aplicación actualiza la información de la tarea en el backend y en la lista de tareas.

Borrado de tareas

1. El usuario selecciona la opción borrar de una tarea.
2. La aplicación confirma pide confirmación para borrar la tarea mediante una alerta.
3. El usuario hace click en la opción de confirmar.
4. La aplicación elimina la tarea del backend y de la lista en la pantalla.

Reglas de negocio

- Las tareas son privadas para cada usuario, un usuario no puede ver las tareas de los demás.
- Los usuarios solo pueden ver, editar y eliminar tareas si tiene una sesión válida activa.
- El tamaño máximo del contenido de las tareas es de 150 caracteres.
- No se pueden crear tareas con un contenido vacío o que incluyan únicamente caracteres de espacio.

Consideraciones generales

- Tanto el backend como el frontend deben estar alojados en una plataforma con servicio

Consideraciones para el backend

- Se deberá validar el acceso a cualquier recurso mediante [OAuth2](#) y [JWT](#), exceptuando el registro de usuarios y el inicio de sesión.
- Debe asumirse que los datos de registro y de inicio de sesión serán cifrados por el frontend.
- Se deberá usar alguna de las siguientes bases de datos:
 - MySQL
 - PostgreSQL
 - MongoDB

Consideraciones para el frontend

- A Cualquier petición realizada al backend se debe añadir autenticación [OAuth2](#) con un bearer token [JWT](#), exceptuando el registro de usuario y el login.
- Los datos para las peticiones al back del registro de usuario y el inicio de sesión se deben cifrar previamente.