

Desarrollar una plataforma web que cumpla los siguientes requerimientos:

Requerimientos funcionales:

- Página de inicio
- Barra de navegación
- Registro de empleados (Tipo documento, documento, Nombres, apellidos, Área y subárea)
 - Las listas desplegables deben estar en la base de datos
 - La Área y Subárea son listas desplegables enlazadas en cascada.
- Listado de empleados con paginación cada 10 registros
- Búsqueda de empleados por documento o Nombre
- Botón para editar el empleado
- Módulo de editar el empleado

Requerimientos técnicos:

- Usar un modelo de arquitectura que separe la vista del controlador a su vez la lógica del controlador y del acceso a datos
- Usar parámetros de seguridad de la información que no permitan tener acceso a información sensible
- Aplique prácticas de desarrollo orientado a objetos
- Aplique el uso de servicios para poder reutilizar el backend en los proyectos de la misma aplicación para móvil
- Si es posible implemente la vista separada del backend
- Realice el desarrollo en Net-Core, Python and Nuxt.js, Vue.js o Node.js

Preguntas teóricas para presentar al ingeniero evaluador

Prepare una presentación de 30 minutos para el equipo técnico evaluador en el cual debe explicar y justificar las siguientes 3 preguntas:

¿El sistema ya compila, pasa las pruebas unitarias y es funcional?

El proyecto tiene un servicio rest el cual tiene el CRUD de todas las tablas.

La interfaz gráfica tiene la pantalla de bienvenida, el listado de empleados, el formulario para agregar empleados.

Lo que faltó fue la parte de edición en la GUI (el servicio lo soporta pero la GUI no).

Una mayor separación de las capas lógicas.

La aplicación está desplegada favor ver el documento anexo (también se encuentra en el repositorio)

¿Si el sistema diseñado anteriormente requiere una alta disponibilidad y concurrencia que componentes propuso para garantizar su disponibilidad y capacidad de respuesta?

Para este caso ha sido desplegado en servicios en la nube para una fácil escalabilidad y rendimiento, desde mi punto de vista es más recomendable con un bajo recurso, son parte de los beneficios del cloud.

¿Explique en sus palabras las capas utilizadas en el modelo anterior y porque es una buena práctica?

La aplicación está dividida en dos partes, por un lado un servicio que corre y tiene el CRUD funcionando.

Aparte se tiene la GUI, se encuentra en el mismo proyecto pero no está amarrado a él por lo que de ser necesario se puede mover la GUI de manera fácil.

Internamente para la parte del backend no pude darle un mejor diseño por cuestiones de tiempo.

Para la parte del front end se le intentó dar un mejor diseño con la ayuda de la librería BackboneJS.

La base de datos es relacional, no hay mayor relevancia en esta capa

¿Explique en sus palabras las buenas practicas de desarrollo que utilizo para construir esta pieza de código fuente?

La separación de las capas permite que un grupo trabaje fácilmente e independiente la interfaz gráfica, la base de datos y el backend ya que esta distribuida incluso se puede alojar en distintos equipos.

se intentó hacer un diseño multicapas pero no lo pude terminar, a su vez se aplicó una lógica orientada a SPA (simple page application).

El recurso que utiliza la interfaz es minimalista por lo que el peso de una petición puede llegar incluso a los 10kb o menos según sea el caso, dando un rápido en la transferencia de la información.