

	Assessment center AREA DE SISTEMAS	
		Versión: 1
		Fecha de emisión: 27/10/2020
		Fecha de actualización: N/A

Nombre: _____

Cargo: _____

Fecha de presentación: _____

Prueba Programador Backend

CONSIDERACIONES GENERALES

Esta prueba está pensada para desarrollarse de manera virtual y para Validar tus conocimientos en el ámbito técnico. Para tener una trazabilidad de tu avance, agrega tags en los commits en donde termines cada punto de la prueba.

Objetivo:

Evaluar los conocimientos y habilidades del candidato para el desarrollo de aplicaciones web a nivel de Backend, debe crear un web services (API) de información de una tabla con su respectiva CRUD y sus respectivas peticiones http.

Lenguaje Programación: para la prueba puede utilizar uno de los siguientes frameworks para Back end PHP(Codeigniter ó Laravel) ó Java, para el Front PLUS Vuejs

Base de Datos: Conocimientos del Lenguaje SQL (CRUD) puede utilizar base de datos sql server, postgresql, mysql.

Puede usar un servidor de aplicaciones como XAMPP o TOMCAT para desplegar proyecto, plus si se expone el proyecto terminado en internet.

Tareas a Realizar

Crear un servicio web o API, que permita a los usuarios autorizados la gestión de sus tareas:

Consultar, Crear, Modificar, Borrar. (Ver Servicios Web

Disponibles) y consumir dichos servicios desde una interfaz gráfica usando JavaScript preferiblemente Vue JS.

Crear las pruebas unitarias para comprobar el correcto funcionamiento de los métodos ofrecidos por el servicio

Documentar los servicios utilizando Swagger o herramienta a su preferencia.

Publicar el código en un repositorio de GitHub, comprimir archivo .zip del proyecto, adjuntar y responder el correo de la prueba en los tiempos determinados a los siguientes correos:

andres.orduz@ospedale.com.co

Alcance:

Se debe incluir:

Modelado de base de datos

Mecanismo de Autenticación y Autorización de Usuarios.

Gestión de usuarios (consultar, crear, actualizar, borrar)

Persistencia de los datos (tareas) en una base de datos tipo SQL.

Métodos que debe publicar el servicio: Consultar, Crear, Actualizar, Borrar Tareas.

Versionar código en GITHUB

Se debe almacenar como mínimo la siguiente información relacionada con las tareas de los usuarios:

Fecha de Creación Descripción,

Estado de la tarea: Se encuentra finalizada (si/no) Fecha de Vencimiento

Autor de la Tarea (Usuario que creó la tarea)

Descripción Métodos del Servicios Web Nombre: "Consultar Tareas"

Método HTTP: GET Path: /tareas/consultar

Parámetros: Los necesarios para que el usuario pueda consultar las tareas programadas por diversos criterios:

Consultar todas las tareas (incluidas las de otros usuarios) Consultar solo mis tareas (las de usuario autenticado)

Consultar solo tareas finalizadas, solo las que están pendientes o todas sin importar su estado de finalización.

Ordenar la consulta por fecha de vencimiento.

Respuesta: Listado de tareas que satisfagan las condiciones establecidas en los parámetros.

Prerrequisito: usuario autenticado

Nombre: Crear de Tarea Método: POST

URL_API: /tareas/crear

Parámetros: debe pedir los parámetros necesarios para almacenar la información sobre la tarea que el usuario está creando.

Respuesta: se devuelve la tarea creada. Prerrequisito: usuario autenticado

Nombre: Actualizar de Tarea Método: POST

URL_API: /tareas/actualizar

Parámetros: debe ser capaz de recibir un objeto con la información que el usuario desea actualizar en una tarea:

Identificador de la tarea // obligatorio

Descripción

Fecha de vencimiento Finalizada (si/no)

Respuesta: se devuelve la tarea actualizada

Prerrequisito: usuario autenticado, usuario autorizado (solo el usuario que ha creado la tarea la puede actualizar)

Borrar de Tarea Método: POST

URL_API: /tareas/borrar

Parámetros: debe recibir información que le permita identificar de manera única la tarea que el usuario desea borrar.

Respuesta: El servicio no devuelve respuesta.

Prerrequisito: usuario autenticado, usuario autorizado (solo el usuario que ha creado la tarea la puede actualizar)

TIPS

La escalabilidad y la eficiencia son importantes, piensa en estrategias para optimizar los recursos del dispositivo.

Piensa en desarrollar productos con muy buena experiencia de usuario y usabilidad además de muy buenos estilos.

Puedes usar patrones de diseño.

Puedes hacer un diseño responsive

Importante que se maneje herramientas para consumir los servicios construidos usando herramientas como Postman

ENTREGABLE

Sube a un repositorio en GitHub la solución del reto.

Debe contener un archivo README.md con una corta explicación del proyecto, los pasos para hacerlo funcionar y alguna otra recomendación que creas relevante.

Debe enviar al correo electrónico andres.orduz@ospedale.com.co el proyecto comprimido en un archivo .zip