

CATEGORÍA: DESARROLLO LIBRE

INTRODUCCIÓN

En este documento se encuentran consignados los lineamientos técnicos, que da la línea a instructores y aprendices - competidores, para reconocer cuales son los requerimientos de la competencia SENASoft 2023, en la categoría de DESARROLLO LIBRE.

Con base en lo aquí expresado se orientarán las rutas de aprendizaje, talleres y conferencias impartidas desde el concurso o preparadas por el equipo de cada regional para mejorar las habilidades de los aprendices. Logrando cubrir las necesidades descritas más adelante en este documento, asumiendo los diversos retos de la competencia, desarrollando las pruebas establecidas y solucionando el problema planteado, generando el producto final.

OBJETIVO

Definir los requisitos de carácter técnico y de conocimientos que regirán la ejecución de las pruebas de la categoría DESARROLLO LIBRE, estableciendo los criterios y competencias a desarrollar, las cuales están asociadas a la formación impartida en el SENA y que se han habilitado para la categoría en el marco del desarrollo de SENASoft 2023.

TEMÁTICA

Diseñar y desarrollar una **aplicación web**, usando alguno los siguientes lenguajes de programación para el lado del servidor (Backend):

- PHP.
- Python.
- Ruby.
- Java Web.
- .NET Core.
- Golang.
- Node JS.
- C#

y del lado del cliente (Frontend) usando los estándares web (HTML5, CSS3 y JS).

Nota:

- Si algún grupo desea utilizar una combinación de los lenguajes anteriormente mencionados para la parte del Backend, también es válido, siempre y cuando hagan uso de una arquitectura bien definida.
- Para el desarrollo del Frontend, se acepta el uso de plantillas y frameworks siempre y cuando sean de uso libre.

REQUERIMIENTOS:

Hardware:

- Procesador Intel Core i5 o superior (arquitectura 64 bits).
- Memoria RAM 8GB mínimo (recomendado 16GB).
- Disco duro 250GB mínimo.

Software:

- SO Windows 10 Pro.
- Paquete AMP:
 - XAMPP PHP 7.2.3 (x64) /WAMP/App Serv/Laragon/IIS.
- Bases de Datos
 - ✓ Relacional (MySQL, MariaDB, Sql Server Express o PostgreSQL).
 - ✓ No relacional (MongoDB, Cassandra, Azure Cosmos DB).
- Herramienta Ofimática Microsoft Office 2010 o superior.
- Editores de Texto:
 - Sublime Text 3.
 - Visual Studio Code.
 - Visual Studio Community.
 - Atom.
- IDE:
 - Netbeans.
 - Eclipse.
- Frameworks Server-Side:
 - Laravel.
 - CodeIgniter.
 - Cake PHP.
 - Symphony.
 - Django.
 - Flask.
 - Ruby on Rails.
 - Struts.
 - Spring Boot.
- Frameworks:
 - Bootstrap.
 - Materialize.
 - Angular.
 - Vue JS.
 - React JS.
 - ASP.NET Core MVC
- Librería JS:
 - JQuery.
- Herramientas de gestión de contenedores:

- Docker Desktop.
- Minikube.
- Herramientas de preferencia para Modelamiento de datos, Arquitectura y diseño de interfaces (UX/UI).

DISTRIBUCIÓN DE TRABAJO DE LA COMPETENCIA

| Jornada | Competencias Relacionadas | Conocimientos |
|---|---|---|
| DÍA 1 ANÁLISIS DE LA INFORMACIÓN Y PLANEACIÓN DEL PROYECTO. DISEÑO DE LA SOLUCIÓN PROPUESTA. | ANALIZAR LOS REQUISITOS DEL CLIENTE PARA CONSTRUIR EL SISTEMA DE INFORMACIÓN. | LEVANTAMIENTO DE INFORMACIÓN ANÁLISIS DE REQUERIMIENTOS DISEÑOS DE FORMULARIOS (MOCKUPS) DISEÑO DE BASES DE DATOS (MR Y DICCIONARIO DE DATOS). |
| DÍA 2 SOLICITUD DE ENLACE DE REPOSITORIO. PRIMERA PRESENTACIÓN ARQUITECTURA Y UX/UI. | DISEÑAR EL SISTEMA DE ACUERDO CON LOS REQUISITOS DEL CLIENTE. | OFD (OBJECT FLOW DIAGRAM). SE PUEDE SOLICITAR ACLARACIONES AL EVALUADOR. |
| DÍA 3 DESARROLLO E IMPLEMENTACIÓN DE LA SOLUCIÓN PROPUESTA. AVANCES DE LA SOLUCIÓN REVISIÓN AL CÓDIGO APOYO DE DUDAS. PRESENTACIÓN FINAL DEL SISTEMA. | CONSTRUIR EL SISTEMA QUE CUMPLA CON LOS REQUISITOS DE LA SOLUCIÓN INFORMÁTICA. APLICAR BUENAS PRÁCTICAS DE CALIDAD EN EL PROCESO DE DESARROLLO DE SOFTWARE, DE ACUERDO CON EL REFERENTE ADOPTADO POR LA EMPRESA. | ESTRUCTURA Y MODELO DE LA SOLUCIÓN PROPUESTA. DESARROLLO DE LOS MÓDULOS DE LÓGICA DE NEGOCIOS DE LA SOLUCIÓN PROPUESTA. MÓDULOS DE MANTENIMIENTO DE DATOS COMUNES (REUTILIZACIÓN DE CÓDIGO) |

EVALUACIÓN

El proceso de evaluación para esta versión SENASoft 2023 en la categoría DESARROLLO LIBRE será liderada por la empresa SIIGO, quienes establecen la prueba a desarrollar y los criterios de evaluación en esta categoría

utilizando sus herramientas, teniendo en cuenta la industria y las competencias desarrolladas por los aprendices SENA habilitados para la misma.

A continuación, se enumeran los ítems que se tendrán en cuenta en el proceso de evaluación por el comité técnico:

- Modelamiento datos.
- Arquitectura y servicios.
- UX/UI.
- Desarrollo en Backend.
- Desarrollo en Frontend.
- Solución óptima al problema.
- Cumplimiento.
- Trabajo en equipo.
- Uso de herramientas de control de versiones.

CRITERIOS DE DESEMPATE

Para definir el equipo ganador, en caso de empate se tomará en su orden los siguientes criterios:

1. **Tiempo de entrega:** se verificará el tiempo de entrega en cada una de las jornadas asignando un punto en cada jornada a la pareja que haya entregado primero su prueba.
2. **Documentación de líneas de código:** si persiste el empate se dará un punto adicional por cada prueba al grupo que utilice comentarios para documentar las líneas de los algoritmos realizados.
3. **Documentación técnica:** si persiste el empate se dará un punto adicional al grupo que haya realizado en mayor detalle la documentación de análisis para la solución de cada prueba.

Reto: Encontrar la ruta más óptima**Descripción del Desafío**

El objetivo de este desafío es desarrollar una aplicación web que permita encontrar la ruta más óptima entre un conjunto de ubicaciones en un mapa, teniendo en cuenta un recorrido específico. Esta aplicación será de gran utilidad en situaciones de rutas de distribución de mercancías, donde se deben entregar pedidos en diferentes puntos desde una ubicación de origen.

Funcionalidades Requeridas**1. Módulo de Carga de datos de prueba (Obligatorio)**

- Desde un archivo plano construido en formato JSON, el sistema debe permitir cargar la información de la ruta de entrega. El archivo define:
 1. Los puntos o ubicaciones de entrega.
 2. Las conexiones bidireccionales entre los puntos y su peso (este peso puede representar distancia en kilómetros o tiempo de recorrido), y
 3. El punto inicial de la ruta.

```
{
  "ubicaciones": [
    {
      "nombre": "A",
      "posX": 20,
      "posY": 20
    },
    {
      "nombre": "B",
      "posX": 45,
      "posY": 60
    },
    {
      "nombre": "C",
      "posX": 79,
      "posY": 90
    },
    {
      "nombre": "D",
      "posX": 56,
      "posY": 79
    }
  ],
  "conexiones": [
    {
      "ubicacion1": "A",
      "ubicacion2": "B",
      "peso": 20
    },
    {
      "ubicacion1": "C",
      "ubicacion2": "D",
      "peso": 50
    }
  ],
  "inicio": "D"
}
```

2. Módulo de Creación de Rutas:

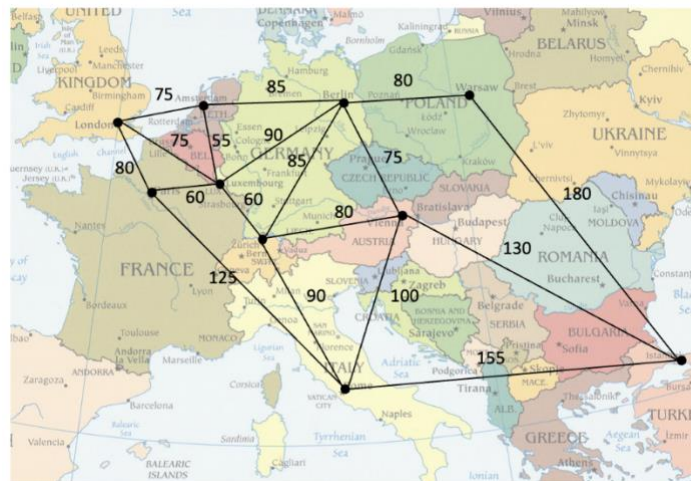
- Los usuarios deben tener la capacidad de crear puntos en un plano cartesiano, especificando sus coordenadas (posiciones X, Y). Los puntos representarán las ubicaciones en las cuales se deben realizar entregas.
- Se debe tener la posibilidad de conectar puntos de manera bidireccional estableciendo un peso o valor de costo de pasar de un punto al otro.
- Poder seleccionar un punto de origen dentro de la lista de puntos de entrega.
- La aplicación debe calcular la ruta más eficiente, de acuerdo a los pesos o costos de conexión entre las ubicaciones, para realizar la entrega de los pedidos desde el punto de origen, teniendo en cuenta todos los puntos de entrega.
- Se puede utilizar un algoritmo de grafos para resolver este problema.

3. Almacenamiento de datos:

- Todas las ubicaciones y rutas se deben poder guardar en una base de datos, para posteriores cambios o nuevos cálculos.

Puntos Adicionales (Opcionales):

- Para obtener puntos adicionales en el desafío, puedes implementar una funcionalidad en el frontend que permita la visualización de las posiciones en un mapa. Puedes utilizar bibliotecas o servicios de mapas, como Google Maps, para mostrar las ubicaciones y trazar la ruta en el mapa.



- Los usuarios deben poder registrarse en la plataforma utilizando su dirección de correo electrónico y contraseña. Debe implementarse un sistema de confirmación mediante un enlace enviado al correo electrónico registrado para validar su cuenta.
- La aplicación debe validar que un usuario no inicie sesión desde dos ubicaciones diferentes al mismo tiempo. Si un usuario inicia sesión desde una segunda ubicación, la primera sesión debe invalidarse automáticamente.

Notas Finales:

Este desafío es una excelente oportunidad para demostrar tus habilidades en desarrollo web, algoritmos de grafos, seguridad, usabilidad, y manejo de sistemas de control de versiones. Se valorará la eficiencia en el cálculo de rutas y la claridad en la presentación de resultados en la interfaz de usuario.

¡Buena suerte en tu desarrollo y que disfrutes resolviendo este desafío de desarrollo de software!