Documentación

***Lenguaje utilizado: C#.***

***Versión del framework: 4.8.***

Se presenta el siguiente documento con el fin de explicar la funcionalidad de cada una de las partes de la aplicación desarrollada.

Contenido

[**1.** **Entidades utilizadas** 2](#_Toc132852253)

[**2.** **Métodos y lógica empleada** 3](#_Toc132852254)

**Entidades utilizadas**

# **BloqueTiempo**

Representa un intervalo de tiempo, indicando un inicio y un fin.

**Propiedades:**

1. **string – Inicio.**
2. **string – Fin.**

# **Persona**

Representa las ocupaciones laborales de una persona.

**Propiedades:**

1. **string – Nombre.**
2. **BloqueTiempo – JornadaLaboral.**
3. **List<BloqueTiempo> – Actividades.**

**Métodos y lógica empleada**

1. **List<Persona> GetPersonas()**

**Método encargado de obtener los datos de prueba.**

Devuelve una instancia de 3 personas con datos pre-establecidos.

1. **List<BloqueTiempo> DevolverHorariosPosibles(List<Persona> Personas, double Duracion)**

**Método principal encargado de obtener el resultado solicitado (horarios disponibles para la reunión de duración indicada).**

Recibe como parámetros la lista de personas y la duración de la reunión.

Devuelve una lista tipo BloqueTiempo con el resultado final, donde serán objetos BloqueTiempo que representen los horarios posibles (y por cada uno su horario limite) para realizar una reunión, según la duración pre-establecida.

1. **List<List<BloqueTiempo>> GetTiempoLibrePersonas(List<Persona> Personas, double Duracion)**

**Obtiene una lista principal que contendrá los horarios libres de cada una de las personas.**

Recibe como parámetros la lista de personas y la duración de la reunión.

Devuelve una lista del tipo List<BloqueTiempo> con todos los tiempos libres, donde cada sub-lista representa el tiempo libre de una de las personas, en base a la duración establecida.

1. **List<BloqueTiempo> GetHorariosDisponibles(Persona Persona, double Duracion)**

**Busca los intervalos de tiempo, en el cual, la persona no se encuentra realizando ninguna actividad, y a su vez, está dentro del margen de la jornada laboral.**

Recibe como parámetros un objeto Persona y la duración de la reunión.

Devuelve una lista del tipo BloqueTiempo con los horarios disponibles del objeto Persona, teniendo en cuenta la duración.

1. **bool VerificarIntervalo (string Inicio, string Fin, double Duracion)**

**Verifica si el intervalo establecido de Inicio a Fin, es como mínimo igual a la duración de la reunión.**

Recibe como parámetros la duración de la reunión, el inicio y el fin de un intervalo.

Devuelve Verdadero si dicho intervalo es mayor o igual a la duración, Falso en caso contrario.

1. **TimeSpan GetHorarioTimeSpan(string Horario)**

**Realiza una conversión de una cadena.**

Recibe como parámetro un string, el cual representa un horario en formato *“HHMM”.*

Devuelve un objeto tipo TimeSpan con el horario equivalente a la cadena recibida.

1. **List<BloqueTiempo> GetMatchHorarios(List<List<BloqueTiempo>> DispPersonas, double Duracion)**

**Método encargado de buscar coincidencias entre los horarios de todas las personas.**

Recibe como parámetros una lista del tipo List<BloqueTiempo> que contendrá los horarios disponibles, agrupados por cada persona, y además, la duración de la reunión.

Devuelve una lista del tipo BloqueTiempo con todos los intervalos donde es posible realizar la reunión.

1. **List<BloqueTiempo> CompararHorarios(List<BloqueTiempo> DispPersona, List<BloqueTiempo> DispMatch, double Duracion)**

**Se encarga de comparar los horarios de una persona contra los horarios de las demás personas.**

Recibe como parámetros una lista del tipo BloqueTiempo que contendrá las disponibilidades de una persona, otra lista del tipo BloqueTiempo que contendrá los horarios a comparar con los de dicha persona y la duración de la reunión.

Devuelve una lista tipo BloqueTiempo con los horarios que coincidieron con un margen mínimo al de la duración.

1. **List<List<BloqueTiempo>> AgruparHorarios(List<BloqueTiempo> DispPersonas, int CantPersonas)**

**Método encargado de reagrupar los horarios coincidentes en sub-grupos, donde cada grupo tendrá un horario de cada persona que haya coincidido.**

Recibe como parámetros una lista tipo BloqueTiempo con los horarios que coincidieron y la cantidad de personas que se evaluaron anteriormente.

Devuelve una lista del tipo List<BloqueTiempo> con los horarios reagrupados.

1. **List<BloqueTiempo> GetHorariosLimites(List<List<BloqueTiempo>> MatchHorarios, double Duracion)**

**Se encarga de buscar el horario límite en el cual se puede iniciar la reunión.**

Recibe como parámetros una lista tipo List<BloqueTiempo> con los horarios, agrupados, que hayan coincidido y la duración de la reunión.

Devuelve una lista tipo BloqueTiempo con los horarios de inicio y límite de cada posible reunión.

1. **BloqueTiempo BuscarHorarioLimite(List<BloqueTiempo> Horarios, double Duracion)**

**Método encargado de buscar un horario límite para un solo conjunto de datos.**

Recibe como parámetros una lista tipo BloqueTiempo con los horarios y la duración de la reunión.