



Especificación

DISEÑO E IMPLEMENTACIÓN DE UN LENGUAJE

v1.0.0

Índice

1. Equipo	3
2. Repositorio	3
3. Dominio	3
4. Construcciones	4
5. Casos de prueba	4
6. Ejemplos	5

1. Equipo

Nombre	Apellido	Legajo	E-mail
Julieta	Techenski	62547	jtechenski@itba.edu.ar
Matias	Coleur	63461	mcoleur@itba.edu.ar
Valentina	Marti Reta	63225	vmartireta@itba.edu.ar

2. Repositorio

La solución y su documentación serán versionadas en: [Flex-Bison-Compiler](#). El repositorio creado: [Repositorio](#).

3. Dominio

Lenguaje declarativo diseñado específicamente para organizar horarios tanto personales como empresariales. El lenguaje facilitará la creación y gestión de usuarios, la asignación de tareas, y la definición de prioridades entre los distintos tipos de usuarios, proporcionando una solución flexible y adaptable para una amplia gama de necesidades de planificación y organización. La salida del compilador será en formato LaTeX/Markdown (a decidir), lo que permitirá generar representaciones visuales de los horarios, como cronogramas semanales, diagramas de Gantt, calendarios y otros tipos de gráficos útiles para la administración del tiempo y los recursos.

La salida en LaTeX o en Markdown permite una alta compatibilidad con herramientas de edición y sistemas de documentación. Por lo tanto, existe la posibilidad de integrar fácilmente las salidas generadas con otros sistemas de gestión de proyectos o calendarios mediante exportaciones automatizadas.

A su vez, permitirá importar y exportar la información en lenguaje JSON con un formato adecuado, por lo que ayudará al uso de bases de datos existentes y permitirá un intercambio fluido de información entre diferentes departamentos o miembros del personal de la empresa.

4. Construcciones

El lenguaje desarrollado facilitará las siguientes construcciones y funcionalidades:

1. Se podrán definir usuarios con atributos personalizados, como nombre, rol, disponibilidad horaria, grupo de trabajo, etc.]
2. Se facilitará la agrupación de usuarios en equipos o departamentos, lo cual es especialmente útil para contextos empresariales.
3. Se proporcionan comandos para la creación de tareas, especificando detalles como el nombre de la tarea, descripción, duración, fecha de inicio y fin, y requisitos previos (Ya sea que se necesita terminar otra tarea previa).
4. Incluirá un sistema para establecer prioridades entre los usuarios, permitiendo decidir quiénes deben realizar ciertas tareas primero en función de su rol, capacidad, o importancia.
5. Soportará reglas de negocio personalizables, como la asignación preferencial de tareas a usuarios específicos o la priorización de tareas.
6. Permitirá al programador poder importar y exportar información en formato JSON para poder ser utilizada en otros programas o para poder ser compartido.
7. Se podrán anidar ciertos funcionamientos ya sea como la creación de usuarios, la asignación de un grupo de tareas o utilización de departamentos de usuarios.
8. Se permitirá la creación de eventos, en donde se le asignan al usuario/o a un grupo/s
9. Se incluirá la posibilidad de generar gráficos que representen las tareas de los usuarios en cuestión. Se podrán utilizar los siguientes gráficos:
 - **Cronogramas Semanales:** Visualizaciones claras de la distribución de tareas y eventos durante una semana.
 - **Diagramas de Gantt:** Muestra la secuencia de tareas y su duración, facilitando la gestión de proyectos.
 - **Calendarios Personalizados:** Representación mensual o diaria de eventos y tareas asignadas.
 - **Tablas de Disponibilidad:** Resúmenes de la disponibilidad de usuarios o recursos a lo largo del tiempo.

5. Casos de prueba

*Se proponen los siguientes casos iniciales de prueba de **aceptación**:*

- (I). *Un programa que permite la creación de usuarios.*
- (II). *Un programa que permite la creación de grupos.*
- (III). *Un programa que permite la creación de tareas.*
- (IV). *Un programa que relacione usuarios con grupos y con tareas.*
- (V). *Un programa que exporte e importe información en este formato para ser utilizada*

*Además, los siguientes casos de prueba de **rechazo**:*

1. *Un programa malformado.*
2. *Un programa que le asigne una tarea a un usuario que no existe.*
3. *Un programa que relaciona tareas con grupos que no existen.*
4. *Un programa que le asigne a un grupo un usuario que no existe.*
5. *Un programa que intente importar o exportar tipos de datos incompatibles.*

6. Ejemplos

Creación de un grupo, un usuario y la asignación del mismo a un grupo.

```
// Creación de un grupo:
group "IT";

// Creación de un usuario y asignación del mismo a un grupo:
user "Lionel Messi" group "IT" role "admin" weekday [Lunes, Viernes] hours
"8-16";
```

Creación de un evento, una tarea y la asignación de las mismas a un usuario o grupo. Se utiliza el "define" para anidar tareas y eventos bajo un nombre para posteriormente generar un formato de salida utilizando ese nombre.

```
define "mis horarios" {

// Crear un evento y asignarlo a un grupo:
event "Mundial" group "Fútbol" start_date 07-09-2024 end_date 20-09-2024
start_time 09:00 end_time 12:00;

// Crear un evento y asignarlo a un usuario:
event "Mundial" user "Lionel Messi" start_date 07-09-2024 end_date
20-09-2024 start_time 09:00 end_time 12:00;

// Crear una task y asignarla a un usuario o grupo:
task "Entrenamiento" user "Lionel Messi" date 7-09-2024 start_time 09:00
end_time 12:00 description "El campeón del mundo tiene que entrenar";

}
```

Generación de la salida de lo definido en el formato elegido y exportación del mismo:

```
// Generar un calendario semanal:
generate "Horario semanal de Lionel Messi" from "mis horarios" type
w_calendar users ["Lionel Messi"] start_date 20-09-2024
```

```
// Generar un calendario mensual:  
generate "Horario mensual de Empresa" from "mis horarios" type m_calendar  
users all start_date 15-09-2024  
  
// Exportación de los calendarios:  
export "Horario Agosto" path "./horarios"  
  
// Importación de calendarios:  
import "Horario Agosto" path "./horarios"
```