



# Especificación

## DISEÑO E IMPLEMENTACIÓN DE UN LENGUAJE

v1.0.0

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

## 1. Equipo

Nombre	Apellido	Legajo	E-mail
Eduardo Federico	Madero Torres	59494	<a href="mailto:emadero@itba.edu.ar">emadero@itba.edu.ar</a>
Juana	Mendonca Gaona	61185	<a href="mailto:jmendonca@itba.edu.ar">jmendonca@itba.edu.ar</a>
Cristian	Tepedino	62830	<a href="mailto:ctepedino@itba.edu.ar">ctepedino@itba.edu.ar</a>
Luca	Bloise	63004	<a href="mailto:lbloise@itba.edu.ar">lbloise@itba.edu.ar</a>

## 2. Repositorio

La solución y su documentación serán versionadas en: [TP-GANTT](https://github.com/juanamendonca/TP-GANTT)  
(<https://github.com/juanamendonca/TP-GANTT>).

### 3. Dominio

Desarrollar un lenguaje que permita crear y graficar diagramas de Gantt. El lenguaje debe permitir crear proyectos y definir su estructura, donde se pueden agregar tareas especificando su duración o inicio y fin, sus dependencias y asignarles un puntaje. Además debe permitir especificar máxima cantidad de tareas o puntos que se puedan realizar en simultáneo. Debe permitir clasificar las distintas tareas en categorías definidas en la estructura del proyecto.

Como resultado, se debe poder obtener un programa en python que utiliza Plotly para ilustrar los diagramas de Gantt. La implementación satisfactoria de este lenguaje permitiría evidenciar diagramas de Gantt con la organización más eficiente de las tareas, según los requerimientos del proyecto.

### 4. Construcciones

El lenguaje desarrollado debería ofrecer las siguientes construcciones, prestaciones y funcionalidades:

- (I). Se podrá crear un *project* con su nombre y tipo de formato (con duración o inicio y fin de tarea).
- (II). El formato del proyecto podrá ser del tipo *hour*, *day*, *week*, *month* o *date* (se especifica inicio y fin de cada tarea).
- (III). Se podrán crear una o varias *task* dentro de cada *project* y cada una tendrá un nombre y características propias.
- (IV). Se podrá especificar las categorías disponibles para cada *task* en la estructura del *project*.
- (V). Se podrá especificar la máxima cantidad de *tasks* que pueden ocurrir en simultáneo.
- (VI). Se podrá especificar la fecha de inicio del *project*.
- (VII). Se podrán agregar las *dependency* entre *tasks*, (qué tareas se deben finalizar para comenzar).
- (VIII). Se le podrán asignar *points* a cada *task*.
- (IX). Se podrá especificar la máxima cantidad de *points* que pueden ocurrir en simultáneo.
- (X). En caso de formato *date* se deberá ingresar *start* y *finish* de cada *task*. En caso contrario el *length*.
- (XI). En caso de definir *categories* se puede asignar una *category* a cada *task*.
- (XII). Se podrá definir una *task* como *unique* para indicar que no se pueden realizar otras tareas en simultáneo.

## 5. Casos de Prueba

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

- (I). Un programa que construya un *project* con formato *hour* y especifique una *task* con su duración.
- (II). Un programa que construya un *project* con formato *date* y especifique una *task* con fecha de inicio y fin.
- (III). Un programa que construya un *project* con formato *week* y con múltiples *task*.
- (IV). Un programa que construya un *project* con formato *day*, categorías específicas, múltiples *task* y asignaciones de esas categorías a las *tasks*.
- (V). Un programa que construya un *project* con formato *day*, categorías específicas, múltiples *task* y asignaciones de esas categorías a las *tasks*.
- (VI). Un programa que construya un *project* con múltiples *task* y *dependencies*.
- (VII). Un programa que construya un *project* con cantidad máxima de *points* en simultáneos, y múltiples *task* con *points* específicos.
- (VIII). Un programa que construya un *project* con formato *month*, múltiples *task* y una *task* específica como *unique*.
- (IX). Un programa que construya un *project* con múltiples *task*, algunas de las cuales tienen *dependencies* entre sí, y establezca una *task* como *unique*.
- (X). Un programa que construya un *project* con múltiples *task*, algunas de las cuales se asignan a categorías específicas, y establezca una cantidad máxima de *points* simultáneos y asignar puntaje a algunas de ellas.

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

- (I). Un programa en dónde no se especifique el formato del *project*.
- (II). Un programa que asigne a una *task* una *category* no definida.
- (III). Un programa que asigne a una *task* *points* mayores a la cantidad máxima de *points* simultáneos.
- (IV). Un programa que asigne una duración negativa a una *task*.
- (V). Un programa que establezca una fecha de inicio posterior a la fecha de finalización para una *task*,

## 6. Ejemplos

Creación de un proyecto con diferentes tareas, casos más simples.

```
// Crear un nuevo proyecto con formato hora y una tarea
project "Proyecto1" format hour {
```

```

task "Tarea1" length 4;
}

// Crear un nuevo proyecto con formato fecha y una tarea
project "Proyecto2" format date {
task "Tarea2" start "2024-04-01" finish "2024-04-05";
}

// Crear un nuevo proyecto con formato week y una múltiples tareas.
project "Proyecto3" format week {
task "Tarea3" length 2;
task "Tarea4" length 3;
task "Tarea5" length 1;
}

// Crear un nuevo proyecto con formato day, categorías específicas y múltiples
tareas.
project "Proyecto4" format day {
categories "Categoría1" "Categoría2" "Categoría3";
task "Tarea6" length 1 category "Categoría1";
task "Tarea7" length 2 category "Categoría2";
}

// Crear un nuevo proyecto con formato month, tareas y dependencias.
project "Proyecto5" format month {
task "Tarea8" length 2;
task "Tarea9" length 3 depends_on "Tarea8";
}

// Crear un nuevo proyecto con formato day, tareas y máximo de tareas a realizar
simultáneamente.
project "Proyecto6" format day {
max_points 10
task "Tarea1" length 2 points 3;
task "Tarea2" length 3 points 4;
task "Tarea3" length 1 points 2;
task "Tarea4" length 4 points 5;
}

// Crear un nuevo proyecto con formato month, tareas y máximo de tareas a
realizar simultáneamente.
project "Proyecto7" format month {
task "Tarea1" length 2;
task "Tarea2" length 3 unique;
task "Tarea3" length 1;
}

// Crear un nuevo proyecto con formato week, tareas, duración, dependencias.
project "ProyectoDependenciasUnica" format week {
task "Configuración de la base de datos" length 2;

```

```
task "Desarrollo del backend" length 3 depends_on "Configuración de la base de
datos";
task "Despliegue del servidor" length 2 depends_on "Desarrollo del backend";
task "Configuración del frontend" length 2 unique;
}
```

Creación de proyectos, con múltiples tareas y características, más complejos.

```
// Crear un programa que construya un proyecto con múltiples tareas,
// asignaciones de categorías, dependencias entre tareas, y asignación de puntos

project "ProyectoComplejo" format day {
max_tasks 5;
start_date "2024-04-01";
categories "Desarrollo" "Pruebas" "Implementación";
task "Diseño de la interfaz" length 2 category "Desarrollo" points 3;
task "Desarrollo del backend" length 3 category "Desarrollo" points 5;
task "Configuración del entorno de pruebas" length 1 category "Pruebas" points
2;
task "Ejecución de pruebas unitarias" length 2 category "Pruebas" points 4
depends_on "Configuración del entorno de pruebas";
task "Implementación de la funcionalidad principal" length 4 category
"Implementación" points 6 depends_on "Desarrollo del backend";
task "Pruebas de integración" length 2 category "Pruebas" points 4 depends_on
"Ejecución de pruebas unitarias" depends_on "Implementación de la funcionalidad
principal";
}
```