



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
Tiziano	Fuchinecco	64191	tfuchinecco@itba.edu.ar
Santiago	Devesa	64223	sdevesa@itba.edu.ar
Nicolas	Revale	64227	nrevale@itba.edu.ar
Tomás	Balboa	64237	tbalboakoolen@itba.edu.ar

2. Repositorio

La solución y su documentación serán versionadas en: [development](#)

3. Dominio

El objetivo es desarrollar un lenguaje que permita crear, organizar y visualizar horarios académicos de manera eficiente. Este lenguaje estará diseñado para definir materias, profesores, horarios y aulas de una institución educativa, y generará como salida un archivo visual (HTML).

Actualmente, la creación de horarios académicos es un proceso manual, sujeto a errores y conflictos como superposición de clases o asignación de docentes a más de un curso en el mismo horario. Un DSL enfocado en la planificación académica permitirá definir reglas y restricciones de manera programática, evitando estos problemas y automatizando la generación de horarios.

La implementación satisfactoria de este lenguaje facilitará la planificación de clases en universidades, colegios o institutos, reduciendo el esfuerzo manual, minimizando errores y mejorando la organización académica.

4. Construcciones

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

Definición de materias, con nombre, código y duración de la clase.

Asignación de profesores, vinculados a materias y horarios específicos.

Definición de aulas, con capacidad máxima y restricciones (ej. laboratorio, auditorio, etc.).

Especificación de horarios, indicando día, hora de inicio y fin de cada clase.

Personalización de Horarios: Permitir intervalos o días libres sin clases.

Reglas de validación para evitar conflictos de horario entre profesores y aulas.

Resolución de superposiciones: Logra encontrar distintas alternativas horarias.

Exportación del horario en formato HTML.

Operadores lógicos y condicionales para restricciones y reglas personalizadas como AND y OR.

Generación automática de horarios con base en la disponibilidad de aulas y profesores.

5. Casos de Prueba

Casos de Aceptación (funcionamiento esperado)

1. Un programa que permita asignar la materia Física III (Aula: 101 | 14:00 -16:00).
2. Un programa que genere horarios para 5 materias.
3. Un programa que asigne un aula específica (Laboratorio) a una materia específica (Física).
4. Un programa que genere un horario completo y lo exporte en HTML.
5. Un programa que detecte y evite la superposición de Física y Química en el mismo horario.
6. Un programa que genere un horario en formato HTML con colores diferenciando materias.
7. Un programa que permita al usuario solicitar los viernes libres.
8. Un programa que permita al usuario solicitar el intervalo de (12:00pm - 14:00pm) de los martes, libre.
9. Un programa que permita al usuario elegir su preferencia para cursar en el turno de la mañana.
10. Un programa que permita diferenciar 2 materias por su código. (Ej: Álgebra: 93.21 y Álgebra: 93.58).

Casos de Rechazo (errores que deben ser detectados)

1. Un programa con una asignación de horario en un errónea (que la clase termine antes de arrancar)
2. Un programa que solapa horarios sin detectar el conflicto.
3. Un programa que intenta generar horarios sin definir materias ni profesores.
4. Un programa que asigne clases en días no laborales (Domingo).
5. Un programa que genere horarios sin respetar restricciones impuestas por el usuario.

6. Ejemplos

Creación de un calendario académico con dos materias, en un aula y horario específico, teniendo en cuenta las preferencias del usuario.

```
// Crear 2 materias con su respectivo nombre y duración.
materia MAT101 { nombre: "Matemática I"; codigo: 93.17 ; duracion: 2h; }
materia FIS202 { nombre: "Física II"; codigo: 94.19 ; duracion: 2h; }

// Instanciar 2 profesores que estarán asignados a las materias creadas antes
profesor JuanP { nombre: "Juan Pérez"; materias: [MAT101]; }
profesor MariaG { nombre: "María Gómez"; materias: [FIS202]; }

// Crear 2 aulas para luego ser asignadas a las clases
aula A101 { nombre: "Aula 101"; capacidad: 40; }
aula A202 { nombre: "Aula 202"; capacidad: 30; restricciones: ["Laboratorio"]; }

// Crear horarios
horario {
  materia: MAT101;
  profesor: JuanP;
  aula: A101;
  dia: Lunes;
  inicio: 10:00;
  fin: 12:00;
}

horario {
  materia: FIS202;
  profesor: MariaG;
  aula: A202;
  dia: Martes;
  inicio: 14:00;
  fin: 16:00;
}

// Crear restricciones/preferencias solicitadas por el usuario.
restriccion usuario {
  dias_libres: [Viernes];
  evitar_horas: [{ dia: Martes, inicio: 12:00, fin: 14:00 }];
}
```