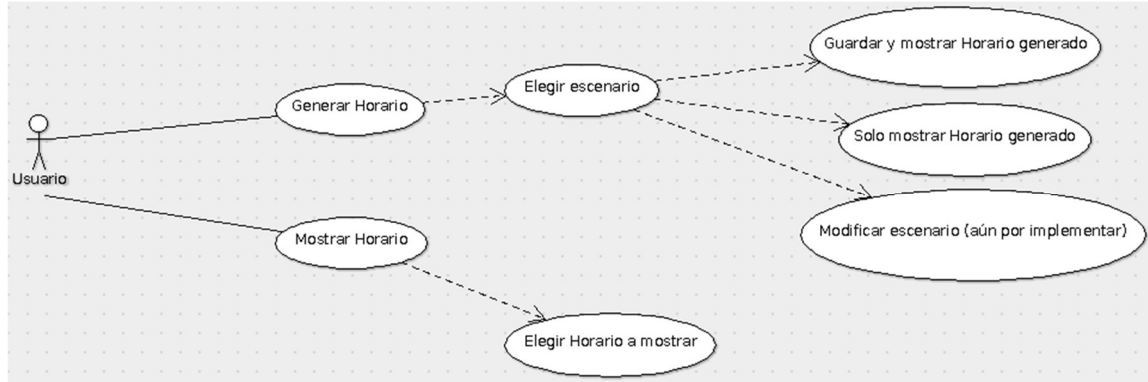


# Generador de horarios PROP G22.5

## 1. Definición de casos de uso

### Diagrama



### Generar Horario

Genera un horario válido en función de un escenario elegido por el usuario.

### Elegir escenario

Se elige un escenario de los escenarios disponibles y lo carga para que ese sea el escenario que se tendrá en cuenta para generar horarios.

### Guardar y mostrar Horario generado

Guarda un horario generado en función de un escenario y lo muestra por pantalla.

### Solo mostrar Horario

Solo se muestra por pantalla un horario generado.

### Modificar Escenario

Permite modificar los datos de un escenario cargado previamente.

### Mostrar Horario

Muestra un horario guardado previamente.

### Elegir Horario a Mostrar

Se elige el horario que se muestra por pantalla de entre todos los horarios guardados disponibles.



- Parámetros:
  - nombre
- addAsignatura: Añade una Asignatura a PlanEstudios.
  - Parámetros:
    - asignatura
- eliminarAsignatura: Elimina una Asignatura a partir de su id.
  - Parámetros:
    - id
- addAula: Añade un Aula a PlanEstudios.
  - Parámetros:
    - aula
- eliminarAula: Elimina un Aula a partir de su id.
  - Parámetros:
    - id
- addRestriccion: Añade una Restriccion a PlanEstudios.
  - Parámetros:
    - restriccion
- eliminarRestriccion: Elimina una Restriccion de PlanEstudios a partir de su posición.
  - Parámetros:
    - posicion
- getNombre: Devuelve el nombre del PlanEstudios.
  - Return: nombre
- getNiveles: Devuelve los Niveles de PlanEstudios.
  - Return: niveles
- getNivel: Devuelve un Nivel de PlanEstudios a partir de su nombre.
  - Parámetros:
    - nombre
  - Return: Nivel con el nombre dado.
- getAsignaturas: Devuelve las Asignaturas de PlanEstudios.
  - Return: asignaturas
- getAsignatura: Devuelve una Asignatura de PlanEstudios a partir de su id.
  - Parámetros:
    - id
  - Return: Asignatura con la id dada.
- getAulas: Devuelve las Aulas de PlanEstudios.
  - Return aulas
- getAulasTeoria: Devuelve las Aulas compatibles con Teoria de PlanEstudios.
  - Return Aulas compatibles con Teoria
- getAulasLaboratorio: Devuelve las Aulas compatibles con Laboratorio de PlanEstudios.
  - Return Aulas compatibles con Laboratorio
- getAulasProblemas: Devuelve las Aulas compatibles con Problemas de PlanEstudios.
  - Return Aulas compatibles con Problemas
- getAula: Devuelve un Aula de PlanEstudios con la id indicada.
  - Parámetros:
    - id
  - Return: Aula con la id indicada

- `getRestricciones`: Devuelve las Restricciones de PlanEstudios.
  - Return: restricciones
- `getRestriccion`: Devuelve la Restriccion que se encuentra en la posicion indicada dentro de restricciones.
  - Parámetros:
    - `posicion`
  - Return: Restriccion que se encuentra en la posicion indicada dentro de restricciones.
- `toString`: Convierte PlanEstudios a String.
  - Return: String con la información que contiene PlanEstudios.

### Nivel: Representa una agrupación entre diversas Asignaturas

- Atributos:
  - `nombre`: nombre que identifica al Nivel (no static).
  - `asignaturas`: conjunto de Asignaturas que pertenecen a Nivel (no static).
  - `restricciones`: conjunto de restricciones que condicionan a Nivel (no static).
- Métodos:
  - `Nivel`: Constructora de la clase Nivel.
    - Parámetros:
      - `nombre`
  - `setNombre`: Asigna un nuevo nombre a Nivel
    - Parámetros:
      - `nombre`
  - `addAsignatura`: Añade una Asignatura a Nivel
    - Parámetros:
      - `asignatura`
  - `eliminarAsignatura`: Elimina una Asignatura de Nivel a partir de su id.
    - Parámetros:
      - `id`
  - `setRestricciones`: Asigna un conjunto de Restricciones a Nivel.
    - Parámetros:
      - `restricciones`
  - `addRestriccion`: Añade una Restriccion a Nivel.
    - Parámetros:
      - `restriccion`
  - `eliminarRestriccion`: Elimina una Restriccion de Nivel a partir de su posición.
    - Parámetros:
      - `posicion`
  - `getNombre`: Devuelve el nombre del Nivel.
    - Return: `nombre`
  - `getAsignaturas`: Devuelve las Asignaturas de Nivel.
    - Return: `asignaturas`
  - `getAsignatura`: Devuelve una Asignatura de Nivel a partir de su id.
    - Parámetros:
      - `id`
    - Return: Asignatura con la id dada.
  - `getRestricciones`: Devuelve las Restricciones de Nivel.
    - Return: `restricciones`
  - `getRestriccion`: Devuelve la Restriccion que se encuentra en la posicion indicada dentro de restricciones.

- Parámetros:
  - posicion
- Return: Restriccion que se encuentra en la posicion indicada dentro de restricciones.
- toString: Convierte Nivel a String.
  - Return: String con la información que contiene Nivel.

### Asignatura: Representa una Asignatura de la que se dan Clases

- Atributos:
  - id: Código identificador de la Asignatura (no static).
  - nombre: Nombre completo de la Asignatura (no static).
  - planEstudios: PlanEstudios al que pertenece la Asignatura (no static).
  - nivel: Nivel al que pertenece la Asignatura (si pertenece a alguno) (no static).
  - sesiones: Sesiones que tiene Asignatura cada semana (no static).
  - grupos: Grupos que tiene la Asignatura (no static).
  - restricciones: Restricciones que afectan a la Asignatura (no static).
- Métodos:
  - Asignatura: Constructora básica de Asignatura.
    - Parámetros:
      - id
      - nom
      - planEstudios
  - Asignatura: Constructora de Asignatura con Nivel.
    - Parámetros:
      - id
      - nom
      - planEstudios
      - nivel
  - Asignatura: Constructora de Asignatura con Sesiones y Grupos.
    - Parámetros:
      - id
      - nom
      - planEstudios
      - sesiones
      - grupos
  - Asignatura: Constructora de Asignatura con Nivel, Sesiones y Grupos.
    - Parámetros:
      - id
      - nom
      - planEstudios
      - nivel
      - sesiones
      - grupos
  - setId: Asigna una nueva id a Asignatura.
    - Parámetros:
      - id
  - setNombre: Asigna un nuevo nombre a Asignatura.
    - Parámetros:
      - nombre
  - setPlanEstudios: Asigna un nuevo PlanEstudios a Asignatura.

- Parámetros:
  - planEstudios
- setNivel: Asigna un nuevo Nivel a Asignatura.
  - Parámetros:
    - Nivel
- setSesiones: Asigna un nuevo conjunto de sesiones a Asignatura.
  - Parámetros:
    - Sesiones
- addSesion: Añade una nueva Sesion a Asignatura.
  - Parámetros:
    - Sesión
- eliminarSesion: Elimina la Sesion de Asignatura que se encuentra en la posicion indicada.
  - Parámetros:
    - posicion
- setGrupos: Asigna un nuevo conjunto de Grupos a Asignatura.
  - Parámetros:
    - grupos
- addGrupo: Añade un nuevo Grupo a Asignatura.
  - Parámetros:
    - grupo
- eliminarGrupo: Elimina un Grupo de Asignatura a partir de su id.
  - Parámetros:
    - Id
- setRestricciones: Asigna un conjunto de Restricciones a Asignatura.
  - Parámetros:
    - restricciones
- addRestriccion: Añade una Restriccion a Asignatura.
  - Parámetros:
    - restriccion
- eliminarRestriccion: Elimina una Restriccion de Asignatura a partir de su posicion.
  - Parámetros:
    - posicion
- getId: Devuelve la id de Asignatura.
  - Return: id
- getNombre: Devuelve el nombre de Asignatura.
  - Return: nombre
- getId: Devuelve la id de Asignatura.
  - Return: id
- getPlanEstudios: Devuelve el PlanEstudios de Asignatura.
  - Return: planEstudios
- tieneNivel: Devuelve si Asignatura pertenece a un Nivel.
  - Return: true si Asignatura pertenece a un Nivel, false en caso contrario.
- getNivel: Devuelve el Nivel de Asignatura.
  - Return: nivel
- getSesiones: Devuelve las Sesiones de Asignatura.
  - Return: sesiones

- getGrupos: Devuelve los Grupos de Asignatura.
  - Return: grupos
- getRestricciones: Devuelve las Restricciones de Asignatura.
  - Return: restricciones
- getAllClases: Devuelve todas las Clases que se pueden generar con las Sesiones y Grupos de Asignatura.
  - Return: Todas las Clases que se pueden generar con las Sesiones y Grupos de Asignatura.
- toString: Convierte Asignatura a String.
  - Return: String con la información que contiene Asignatura.

#### Aula: Representa un Aula donde se realizan Clases.

- Atributos:
  - id: Código identificador de Aula.
  - plazas: plazas que puede alojar el Aula.
  - tipos: TipoClases con los que es compatible el Aula.
- Métodos:
  - Aula: Constructora de la clase Aula.
    - Parámetros:
      - id
      - plazas
      - tipos
  - setId: Asigna una nueva id a Aula.
    - Parámetros:
      - id
  - setPlazas: Asigna un nuevo valor a plazas de Aula.
    - Parámetros:
      - plazas
  - setTipos: Asigna un nuevo conjunto de TiposClase a Aula.
    - Parámetros:
      - Tipos
  - getId: Devuelve la id de Aula.
    - Return: id
  - getPlazas: Devuelve las plazas de Aula.
    - Return: id
  - getTipos: Devuelve los tipos de Aula.
    - Return: tipos
  - tieneTeoria: Devuelve si Aula puede alojar Clases de Teoria.
    - Return: true si Aula puede alojar Clases de Teoria, false en caso contrario.
  - tieneLaboratorio: Devuelve si Aula puede alojar Clases de Laboratorio.
    - Return: true si Aula puede alojar Clases de Laboratorio, false en caso contrario.
  - tieneProblemas: Devuelve si Aula puede alojar Clases de Problemas.
    - Return: true si Aula puede alojar Clases de Problemas, false en caso contrario.
  - toString: Convierte Aula a String.
    - Return: String con la información que contiene Aula.

#### Sesion: Representa una las veces que un SubGrupo da Clase.

- Atributos:

- duracion: Tiempo en horas que dura la Sesión.
- tipo: TipoClase de la Sesión.
- asignatura: Asignatura a la que pertenece la Sesión.
- Métodos:
  - Sesión: Constructora de la clase Sesión.
    - Parámetros:
      - duracion
      - tipo
      - asignatura
  - setDuracion: Asigna un nuevo valor a duracion de Sesión.
    - Parámetros:
      - duracion
  - setTipo: Asigna un nuevo TipoClase a Sesión.
    - Parámetros:
      - tipo
  - setAsignatura: Asigna una nueva Asignatura a Sesión.
    - Parámetros:
      - asignatura
  - getDuracion: Devuelve el valor de duracion.
    - Return: duracion
  - getTipo: Devuelve el TipoClase de Sesión.
    - Return: tipo
  - getAsignatura: Devuelve la Asignatura a la que pertenece Sesión.
    - Return: asignatura
  - toString: Convierte Sesión a un String que contiene su información.
    - Return: String con la información de Sesión.

Grupo: Representa un conjunto de SubGrupos que realizan Clases.

- Atributos:
  - id: código identificador del Grupo.
  - Asignatura: Asignatura a la que pertenece el Grupo.
  - subGrupos: Conjunto de SubGrupos que pertenecen a Grupo.
- Métodos:
  - Grupo: Constructora de la clase Grupo.
    - Parámetros:
      - id
      - asignatura
  - setId: Asigna una nueva id a Grupo.
    - Parámetros:
      - id
  - setAsignatura: Asigna una nueva Asignatura a Grupo.
    - Parámetros:
      - Asignatura
  - setSubGrupos: Asigna un nuevo conjunto de SubGrupos a Grupo.
    - Parámetros:
      - subGrupos
  - addSubGrupo: Añade un nuevo SubGrupo a Grupo.
    - Parámetros:
      - subGrupo
  - eliminarSubGrupo: Elimina el SubGrupo de Grupo con la key indicada.



- Parámetros:
  - Key
- getId: Devuelve la id de Grupo.
  - Return: id
- getAsignatura: Devuelve la Asignatura a la que pertenece Grupo.
  - Return: asignatura
- getSubGrupos: Devuelve los SubGrupos de Grupo.
  - Return: subGrupos.
- getSubGruposTeoria: Devuelve los SubGrupos de Teoria de Grupo.
  - Return: SubGrupos de Grupo de Teoria.
- getSubGruposLaboratorio: Devuelve los SubGrupos de Laboratorio de Grupo.
  - Return: SubGrupos de Grupo de Laboratorio.
- getSubGruposProblemas: Devuelve los SubGrupos de Problemas de Grupo.
  - Return: SubGrupos de Grupo de Problemas.
- getSubGrupo: Devuelve un SubGrupo de Grupo dada su key.
  - Parámetros:
    - key
  - Return: Grupo con la key indicada.
- tieneNivel: indica si el Grupo pertenece a una Asignatura con Nivel.
  - Return: true si el Grupo pertenece a una Asignatura con Nivel, false en caso contrario.
- getNivel: Devuelve el Nivel al que pertenece la Asignatura a la que pertenece Grupo.
  - Return: Nivel al que pertenece la Asignatura a la que pertenece Grupo.
- toString: Convierte Grupo a un String que contiene toda su información.
  - Return: String que contiene la información de Grupo.

**SubGrupo: Representa un conjunto que participa en Clases.**

- Atributos:
  - id: Código identificador del SubGrupo.
  - plazas: Número de alumnos que caben en el SubGrupo.
  - tipo: TipoClase del SubGrupo.
  - grupo: Grupo al que pertenece SubGrupo.
- Métodos:
  - SubGrupo: Constructora de la clase SubGrupo.
    - Parámetros:
      - id
      - plazas
      - tipo
      - grupo
  - setId: Asigna una nueva id a SubGrupo.
    - Parámetros:
      - id
  - setPlazas: Asigna un nuevo valor a plazas.
    - Parámetros:
      - Plazas
  - setGrupo: Asigna un nuevo valor a Grupo.
    - Parámetros:
      - Grupo
  - getId: Devuelve la id del SubGrupo.

- Return: id
- getKey: Genera una key única para el SubGrupo.
  - Return: key única para el SubGrupo.
- getIdCompleta: Devuelve una id hecha con la id del Grupo al que pertenece SubGrupo y la id SubGrupo.
  - Return: id hecha con la id del Grupo al que pertenece SubGrupo y la id SubGrupo.
- getPlazas: Devuelve el valor de plazas de SubGrupo.
  - Return: plazas
- getTipo: Devuelve el TipoClase del SubGrupo.
  - Return: tipo
- getGrupo: Devuelve el Grupo al que pertenece SubGrupo.
  - Return: grupo
- getAsignatura: Devuelve la Asignatura a la que pertenece el Grupo al que pertenece SubGrupo.
  - Return: Asignatura a la que pertenece el Grupo al que pertenece SubGrupo.
- tieneNivel: indica si el SubGrupo pertenece a un Grupo que pertenece a una Asignatura con Nivel.
  - Return: true si el SubGrupo pertenece a un Grupo que pertenece a una Asignatura con Nivel, false en caso contrario.
- getNivel: Devuelve el Nivel al que pertenece la Asignatura a la que pertenece Grupo al que pertenece SubGrupo.
  - Return: Nivel al que pertenece la Asignatura a la que pertenece Grupo al que pertenece SubGrupo.
- toString: Convierte SubGrupo a un String que contiene toda su información.
  - Return: String que contiene la información de SubGrupo.

Clase: Representa una asociación entre un SubGrupo y una Sesion de un mismo TipoClase.

- Atributos:
  - subGrupo: SubGrupo que participa en la Clase.
  - sesion: Sesion que se realiza en la Clase.
- Métodos:
  - Clase: Constructora de la clase Clase.
    - Parámetros:
      - subGrupo
      - sesión
  - setSubGrupo: Asigna un nuevo SubGrupo a Clase.
    - Parámetros:
      - subgrupo
  - setSesion: Asigna una nueva Sesion a Clase.
    - Parámetros:
      - Sesión
  - comprobarRestricciones: Comprueba las Restricciones que afectan a la Clase dados un día, horaIni y Ocupaciones determinados.
    - Parámetro:
      - dia
      - horaIni
      - ocupaciones

- toStringResumido: Convierte Clase a un String que contiene su información de forma resumida.
  - Return: String que contiene la información de Clase de forma resumida.
- toString: Convierte Clase a un String que contiene su información.
  - Return: String que contiene la información de Clase.

Asignacion: Asociación entre una Clase y una Aula en un con una horalNi i diaSemana determinados.

- Atributos:
  - horalNi: hora de inicio de la Asignacion.
  - diaSemana: día de la semana de la Asignacion.
  - aula: Aula donde se produce la Asignacion.
  - clase: Clase que se produce en la Asignacion.
- Métodos:
  - Asignacion: Constructora de la clase Asignacion.
    - Parámetros:
      - horalNi
      - diaSemana
      - aula
      - clase
  - Asignacion: Crea una copia de otra Asignacion.
    - Parámetros:
      - oldAsignacion
  - setHoralNi: Asigna una nueva horalNi a Asignacion.
    - Parámetros:
      - horalNi
  - setDiaSemana: Asigna un nuevo valor a diaSemana de Asignacion.
    - Parámetros:
      - diaSemana
  - setAula: Asigna una nueva Aula a Asignacion.
    - Parámetros:
      - aula
  - setClase: Asigna una nueva Clase a Asignacion.
    - Parámetros:
      - Clase
  - comprobarRestricciones: Comprueba que se cumplen todas las Restricciones que afectan a Asigancion dadas unas Ocupaciones.
    - Parámetros:
      - ocupaciones
    - Return: true si se cumplen todas las Restricciones que afectan a Asigancion dadas unas Ocupaciones, false en caso contrario.
  - noCabeSubGrupo: Comprueba si el SubGrupo de Asignacion no cabe en el Aula de Asignacion.
    - Return: true si el SubGrupo de Asignacion no cabe en el Aula de Asignacion, false en caso contrario.
  - generateKey: Genera una key única para la Asignacion.
    - Return: key única para la Asignacion.
  - getHoralNi: Devuelve la horalNi de Asignacion.
    - Return: horalNi

- `getDiaSemana`: Devuelve el `diaSemana` de la `Asignacion`.
  - `Return`: `diaSemana`
- `getHoraFin`: Devuelve la hora resultante de sumar `horalni` a la `duracion` de clase.
  - `Return`: suma de `horalni` i la `duracion` de clase.
- `getAula`: Devuelve el `Aula` de `Asignacion`.
  - `Return`: `aula`
- `getGrupo`: Devuelve el `Grupo` del `SubGrupo` de la `Clase` de `Asignacion`.
  - `Return`: `Grupo` del `SubGrupo` de la `Clase` de `Asignacion`.
- `getSubGrupo`: Devuelve el `SubGrupo` de la `Clase` de `Asignacion`.
  - `Return`: `SubGrupo` de la `Clase` de `Asignacion`.
- `getClase`: Devuelve la `Clase` de `Asignacion`.
  - `Return`: `clase`
- `getRestricciones`: Devuelve las `Restricciones` que afectan a la `Restriccion`.
  - `Return`: `Restricciones` que afectan a `Asignacion`.
- `getAsignatura`: Devuelve la `Asignatura` del `Grupo` del `SubGrupo` de la `Clase` de `Asignacion`.
  - `Return`: `Asignatura` del `Grupo` del `SubGrupo` de la `Clase` de `Asignacion`.
- `tieneNivel`: Devuelve si la `Asignatura` del `Grupo` del `SubGrupo` de la `Clase` de `Asignacion` tiene `Nivel`.
  - `Return`: `true` si la `Asignatura` del `Grupo` del `SubGrupo` de la `Clase` de `Asignacion` tiene `Nivel`, `false` en caso contrario.
- `getNivel`: Devuelve el `Nivel` de la `Asignatura` del `Grupo` del `SubGrupo` de la `Clase` de `Asignacion`.
  - `Return`: `Nivel` de la `Asignatura` del `Grupo` del `SubGrupo` de la `Clase` de `Asignacion`.
- `toStringCompleto`: Convierte `Asignacion` a un `String` con su información detallada.
  - `String` con la información detallada de `Asignacion`.
- `toString`: Convierte `Asignacion` a un `String` con su información.
  - `String` con la información de `Asignacion`.

**Hora:** Mantiene constancia de lo que ocurre en cada una de las horas de un `Dia`.

- **Atributos:**
  - `niveles`: Niveles de las `Asignaciones` que suceden en `Hora`.
  - `asignaturas`: `Asignaturas` de las `Asignaciones` que suceden en `Hora`.
  - `aulas`: `Aulas` de las `Asignaciones` que suceden en `Hora`.
  - `grupos`: `Grupos` de las `Asignaciones` que suceden en `Hora`.
  - `subGrupos`: `SubGrupos` de las `Asignaciones` que suceden en `Hora`.
  - `asignaciones`: `Asignaciones` que suceden en una `Hora`.
- **Métodos:**
  - `Hora`: Constructora básica de `Hora`
  - `Hora`: Constructora de `Hora` que copia los datos de otra `Hora`.
    - `Parámetros`:
      - `oldHora`
  - `setAsignaciones`: Asigna un nuevo conjunto de `Asignaciones` a `Hora`.
    - `Parámetros`:
      - `asignaciones`
  - `addAsignacion`: Añade una `Asignacion` a `Hora`.
    - `Parámetros`:

- Nivel
- eliminarAsignacion: Dada su key, elimina un Asignacion de Hora.
  - Parámetros:
    - key
- getNiveles: Devuelve los Niveles de las Asignaciones que se dan en Hora.
  - Return: niveles
- tieneNivel: Dado un Nivel devuelve si ese Nivel está en niveles.
  - Parámetros
    - nivel
  - Return: true si nivel está en niveles.
- getNivel: Dado su nombre devuelve un Nivel que se encuentra en niveles.
  - Parámetros:
    - nombre
  - Return: Nivel con el nombre indicado.
- getAsignaturas: Devuelve las Asignaturas de las Asignaciones que se dan en Hora.
  - Return: asignaturas
- tieneAsignatura: Dada una Asignatura devuelve si esa Asignatura está en asignaturas.
  - Parámetros
    - asignatura
  - Return: true si asignatura está en asignaturas.
- getAsignatura: Dado su id devuelve una Asignatura que se encuentra en asignaturas.
  - Parámetros:
    - id
  - Return: Asignatura con el id indicado.
- getAulas: Devuelve las Aulas de las Asignaciones que se dan en Hora.
  - Return: aulas
- tieneAula: Dada una Aula devuelve si esa Aula está en aulas.
  - Parámetros
    - aula
  - Return: true si aula está en aulas.
- getAula: Dado su id devuelve una Aula que se encuentra en aulas.
  - Parámetros:
    - id
  - Return: Aula con el id indicado.
- getGrupos: Devuelve los Grupos de las Asignaciones que se dan en Hora.
  - Return: grupos
- tieneGrupo: Dado un Grupo devuelve si ese Grupo está en grupos.
  - Parámetros
    - grupo
  - Return: true si grupo está en grupos.
- getGrupo: Dado su key devuelve un Grupo que se encuentra en grupos.
  - Parámetros:
    - key
  - Return: Grupo con la key indicada.
- getSubGrupos: Devuelve los SubGrupos de las Asignaciones que se dan en Hora.

- Return: subGrupos
- tieneSubGrupo: Dado un SubGrupo devuelve si ese SubGrupo está en subGrupos.
  - Parámetros
    - grupo
  - Return: true si grupo está en subGrupos.
- getAsignaciones: Devuelve las Asignaciones que suceden en Hora.
  - Return: asignaciones
- getAsignacion: Dada una key devuelve la Asignacion con esa key.
  - Return: Asignacion con la key indicada.

Dia: Mantiene constancia de lo que ocurre en cada uno de los días de Ocupaciones.

- Atributos:
  - niveles: Niveles de las Asignaciones que suceden en Dia.
  - asignaturas: Asignaturas de las Asignaciones que suceden en Dia.
  - aulas: Aulas de las Asignaciones que suceden en Dia.
  - grupos: Grupos de las Asignaciones que suceden en Dia.
  - subGrupos: SubGrupos de las Asignaciones que suceden en Dia.
  - horas: Array de 24 Horas que guardan información de lo que ocurre en cada una de las horas del Dia.
- Métodos:
  - Dia: Constructora básica de Dia
  - Dia: Constructora de Dia que copia los datos de otro Dia.
    - Parámetros:
      - oldDia
  - getNiveles: Devuelve los Niveles de las Asignaciones que se dan en Dia.
    - Return: niveles
  - tieneNivel: Dado un Nivel devuelve si ese Nivel está en niveles.
    - Parámetros
      - nivel
    - Return: true si nivel está en niveles.
  - getNivel: Dado su nombre devuelve un Nivel que se encuentra en niveles.
    - Parámetros:
      - nombre
    - Return: Nivel con el nombre indicado.
  - getAsignaturas: Devuelve las Asignaturas de las Asignaciones que se dan en Dia.
    - Return: asignaturas
  - tieneAsignatura: Dada una Asignatura devuelve si esa Asignatura está en asignaturas.
    - Parámetros
      - asignatura
    - Return: true si asignatura está en asignaturas.
  - getAsignatura: Dado su id devuelve una Asignatura que se encuentra en asignaturas.
    - Parámetros:
      - id
    - Return: Asignatura con el id indicado.
  - getAulas: Devuelve las Aulas de las Asignaciones que se dan en Dia.

- Return: aulas
- tieneAula: Dada una Aula devuelve si esa Aula está en aulas.
  - Parámetros
    - aula
  - Return: true si aula está en aulas.
- getAula: Dado su id devuelve una Aula que se encuentra en aulas.
  - Parámetros:
    - id
  - Return: Aula con el id indicado.
- getGrupos: Devuelve los Grupos de las Asignaciones que se dan en Dia.
  - Return: grupos
- tieneGrupo: Dado un Grupo devuelve si ese Grupo está en grupos.
  - Parámetros
    - grupo
  - Return: true si grupo está en grupos.
- getGrupo: Dado su key devuelve un Grupo que se encuentra en grupos.
  - Parámetros:
    - key
  - Return: Grupo con la key indicada.
- getSubGrupos: Devuelve los SubGrupos de las Asignaciones que se dan en Dia.
  - Return: subGrupos
- tieneSubGrupo: Dado un SubGrupo devuelve si ese SubGrupo está en subGrupos.
  - Parámetros
    - grupo
  - Return: true si grupo está en subGrupos.
- getHoras: Devuelve el Array con las Horas de Dia.
  - Return: horas
- getHora: Devuelve la Hora con la información de la hora indicada.
  - Parámetro:
    - hora
  - Return: Hora con la información de la hora indicada.

Ocupaciones: Guarda la información referente a lo que ocurre en una semana.

- Atributos:
  - dias: Array con los 7 Dias de la semana.
- Métodos:
  - Ocupaciones: Constructora básica de Ocupaciones.
  - Ocupaciones: Constructora de Ocupaciones que copia los datos de otra Ocupaciones.
    - Parámetros: oldOcupaciones.
  - addAsignacion: Añade una Asignacion a Ocupaciones.
    - Parámetros:
      - Asignación
  - getDias: Devuelve el Array con los Dias de ocupaciones.
    - Return: dias
  - getDia: Devuelve el Dia indicado de Ocupaciones
    - Parámetros:
      - dia
    - Return: Dia indicado por dia.

- getHora: Devuelve la Hora correspondiente al día y hora indicados.
  - Parámetros:
    - día
    - hora
  - Return: Hora correspondiente al día y hora indicados.
- toString: Convierte Ocupaciones a un String con toda su información.
  - Return: String con toda la información de Ocupaciones

#### Horario: Gestiona la información de Ocupaciones.

- Atributos:
  - id: Código identificador de Horario.
  - ocupaciones: Conjunto de Asignaciones asociadas a Dias i Horas.
- Métodos:
  - Horario: Constructora básica de Horario.
    - Parámetros:
      - id
  - Horario: Constructora de Horario con Ocupaciones.
    - Parámetros:
      - Id
      - Ocupaciones
  - setId: Asigna una nueva id a Horario.
    - Parámetros:
      - id
  - setOcupaciones: Asigna nuevas Ocupaciones a Horario.
    - Parámetros:
      - Ocupaciones
  - getId: Devuelve la id de Horario.
    - Return: id
  - getOcupaciones: Devuelve las Ocupaciones asociadas a Horario.
    - Return: ocupaciones
  - getDia: Devuelve el Dia indicado de ocupaciones.
    - Parámetros:
      - día
    - Return: Dia indicado de ocupaciones.
  - getHora: Devuelve la Hora indicada del Dia indicado de ocupaciones.
    - Parámetros:
      - día
      - hora
    - Return: Hora indicada del Dia indicado de ocupaciones.
  - toString: Convierte Horario en un String que contiene toda su información.
    - Return: String con toda la información de Horario.

#### Restricción (Clase Abstracta): Representa una Restriccion que condiciona la generación de Horarios.

- Métodos:
  - Restriccion: Constructora básica de Restriccion.
  - getTipoRestriccion: Devuelve el TipoRestriccion de la Restriccion sobre la que se ejecuta.
    - Return: TipoRestriccion de la Restriccion



- comprobarRestriccion: Comprueba que la Restriccion se cumple dada una Clase, el dia y horalni con el que se quieren asignar y las Ocupaciones a las que se quiere asignar.
  - Parámetros:
    - clase
    - horalni
    - dia
    - ocupaciones

**DiaLibre (extends Restriccion):** Contiene información sobre un dia en el que no se da Clase.

- Atributos:
  - dia: Dia de la semana en el que no se da Clase.
- Métodos:
  - DiaLibre: Constructora de la clase DiaLibre.
    - Parámetros:
      - dia
  - setDia: Asigna un nuevo dia a DiaLibre.
    - Parámetros:
      - Dia
  - getDia: Devuelve el dia que no se da Clase.
    - Return dia
  - getTipoRestriccion: Devuelve el TipoRestriccion de la Restriccion sobre la que se ejecuta.
    - Return: TipoRestriccion.DiaLibre
  - comprobarRestriccion: Comprueba que la Restriccion se cumple dada una Clase, el dia y horalni con el que se quieren asignar y las Ocupaciones a las que se quiere asignar.
    - Parámetros:
      - clase
      - horalni
      - dia
      - ocupaciones

**Franja Trabajo (extends Restriccion):** Delimita entre que Horas se debe dar Clase.

- Atributos:
  - horalni: Hora a partir de la cual se puede dar Clase.
  - horaFin: Hora hasta la cual se puede dar Clase.
- Métodos:
  - FranjaTrabajo: Creadora de la clase FranjaTrabajo.
    - Parámetros:
      - horalni
      - horaFin
  - setHoralni: Asigna una nueva horalni a la Restriccion.
    - Parámetros:
      - horalni
  - setHoraFin: Asigna una nueva horaFin a la Restriccion.
    - Parámetros:
      - horaFin

- getHoralNi: Devuelve el valor de horaNi.
  - Return: horaNi
- getHoraFin: Devuelve el valor de horaFin.
  - Return: horaFin
- getTipoRestriccion: Devuelve el TipoRestriccion de la Restriccion sobre la que se ejecuta.
  - Return: TipoRestriccion.FranjaTrabajo
- comprobarRestriccion: Comprueba que la Restriccion se cumple dada una Clase, el dia y horaNi con el que se quieren asignar y las Ocupaciones a las que se quiere asignar.
  - Parámetros:
    - clase
    - horaNi
    - dia
    - ocupaciones

Franja Nivel (extends Restriccion): Delimita entre que Horas se debe dar Clase de Asignaturas de un Nivel.

- Atributos:
  - nivel: Nivel al que se aplica la Restriccion.
  - horaNi: Hora a partir de la cual se puede dar Clase.
  - horaFin: Hora hasta la cual se puede dar Clase.
- Métodos:
  - FranjaNivel: Creadora de la clase FranjaNivel.
    - Parámetros:
      - nivel
      - horaNi
      - horaFin
  - setNivel: Asigna un nuevo Nivel a la Restriccion.
    - Parámetros:
      - nivel
  - setHoralNi: Asigna una nueva horaNi a la Restriccion.
    - Parámetros:
      - horaNi
  - setHoraFin: Asigna una nueva horaFin a la Restriccion.
    - Parámetros:
      - horaFin
  - getNivel: Devuelve el Nivel de la Restriccion.
    - Return: nivel
  - getHoralNi: Devuelve el valor de horaNi.
    - Return: horaNi
  - getHoraFin: Devuelve el valor de horaFin.
    - Return: horaFin
  - getTipoRestriccion: Devuelve el TipoRestriccion de la Restriccion sobre la que se ejecuta.
    - Return: TipoRestriccion.FranjaNivel
  - comprobarRestriccion: Comprueba que la Restriccion se cumple dada una Clase, el dia y horaNi con el que se quieren asignar y las Ocupaciones a las que se quiere asignar.
    - Parámetros:

- clase
- horaNi
- dia
- ocupaciones

Franja Asignatura (extends Restriccion): Delimita entre que Horas se debe dar Clase de una Asignatura.

- Atributos:
  - asignatura: Asignatura a la que se aplica la Restriccion.
  - horaNi: Hora a partir de la cual se puede dar Clase.
  - horaFin: Hora hasta la cual se puede dar Clase.
- Métodos:
  - FranjaAsignatura: Creadora de la clase FranjaAsignatura.
    - Parámetros:
      - asignatura
      - horaNi
      - horaFin
  - setAsignatura: Asigna una nueva Asignatura a la Restriccion.
    - Parámetros:
      - asignatura
  - setHoraNi: Asigna una nueva horaNi a la Restriccion.
    - Parámetros:
      - horaNi
  - setHoraFin: Asigna una nueva horaFin a la Restriccion.
    - Parámetros:
      - horaFin
  - getAsignatura: Devuelve la Asignatura de la Restriccion.
    - Return: asignatura
  - getHoraNi: Devuelve el valor de horaNi.
    - Return: horaNi
  - getHoraFin: Devuelve el valor de horaFin.
    - Return: horaFin
  - getTipoRestriccion: Devuelve el TipoRestriccion de la Restriccion sobre la que se ejecuta.
    - Return: TipoRestriccion.FranjaAsignatura
  - comprobarRestriccion: Comprueba que la Restriccion se cumple dada una Clase, el dia y horaNi con el que se quieren asignar y las Ocupaciones a las que se quiere asignar.
    - Parámetros:
      - clase
      - horaNi
      - dia
      - ocupaciones

NivelHora (extends Restriccion): Restriccion que no pueden solaparse en el tiempo Clases de una Asignaturas de un Nivel que tengan Grupos con la misma id.

- Atributos:
  - nivel: Nivel al que se aplica la Restriccion.
- Métodos:

- NivelHora: Constructora de la clase NivelHora.
  - Parámetros:
    - Nivel
- setNivel: Asigna un nuevo Nivel a la Restriccion.
  - Parámetros:
    - nivel
- getNivel: Obtiene el Nivel de la Restriccion.
  - Return: nivel
- getTipoRestriccion: Devuelve el TipoRestriccion de la Restriccion sobre la que se ejecuta.
  - Return: TipoRestriccion.NivelHora
- comprobarRestriccion: Comprueba que la Restriccion se cumple dada una Clase, el dia y horalni con el que se quieren asignar y las Ocupaciones a las que se quiere asignar.
  - Parámetros:
    - clase
    - horalni
    - dia
    - ocupaciones

Prerrequisito: Indica la existencia de un Prerrequisito entre asignaturas.

- Atributos:
  - asignatura: Asignatura que tiene el Prerrequisito.
  - prerrequisito: Asignatura que es Prerrequisito.
- Métodos:
  - Prerrequisito: Constructora de la clase Prerrequisito.
    - Parámetros:
      - asignatura
      - prerrequisito
  - setAsignatura: Asigna una nueva Asignatura a asignatura.
    - Parámetros:
      - asignatura
  - setPrerrequisito: Asigna una nueva Asignatura a Prerrequisito.
    - Parámetros:
      - Prerrequisito
  - getAsignatura: Devuelve la Asignatura que tiene el prerrequisito.
    - Return: Asignatura que tiene el prerrequisito.
  - getPrerrequisito: Devuelve la Asignatura que es Prerrequisito.
    - Return: Asignatura que es Prerrequisito.
  - getTipoRestriccion: Devuelve el TipoRestriccion de la Restriccion sobre la que se ejecuta.
    - Return: TipoRestriccion.Prerrequisito
  - comprobarRestriccion: Comprueba que la Restriccion se cumple dada una Clase, el dia y horalni con el que se quieren asignar y las Ocupaciones a las que se quiere asignar.
    - Parámetros:
      - clase
      - horalni
      - dia
      - ocupaciones

Correquisito: Indica la existencia de un Correquisito entre asignaturas.

- Atributos:
  - asignatura1: Asignatura que tiene Correquisito.
  - asignatura2: Asignatura que tiene Correquisito.
- Métodos:
  - Correquisito: Constructora de la clase Correquisito.
    - Parámetros:
      - asignatura1
      - asignatura2
  - setAsignatura1: Asigna una nueva Asignatura a asignatura1.
    - Parámetros:
      - asignatura1
  - setAsignatura2: Asigna una nueva Asignatura a asignatura2.
    - Parámetros:
      - asignatura2
  - setPrerrequisito: Asigna una nueva Asignatura a Prerrequisito.
    - Parámetros:
      - Prerrequisito
  - tieneAsignatura: Devuelve si asignatura forma parte de la Restriccion.
    - Parámetros:
      - asignatura
    - Return: true si asignatura forma parte de la Restriccion, false en caso contrario.
  - getIndex: Devuelve si asignatura es asignatura1 o asignatura2.
    - Return: 1 si asignatura es asignatura1, 2 en caso contrario.
  - getTipoRestriccion: Devuelve el TipoRestriccion de la Restriccion sobre la que se ejecuta.
    - Return: TipoRestriccion.Correquisito
  - comprobarRestriccion: Comprueba que la Restriccion se cumple dada una Clase, el dia y horalni con el que se quieren asignar y las Ocupaciones a las que se quiere asignar.
    - Parámetros:
      - clase
      - horalni
      - dia
      - ocupaciones
  - toString: Convierte Correquisito a un String que contenga su información.
    - Return: String con la información de Correquisito.

### 3. Descripción de estructuras de datos y algoritmos utilizado para implementar las funcionalidades principales (generación de algoritmos)

#### Estructuras de datos

- ReturnSet: Se usa para devolver valores de diferente tipo en las funciones de generación de horarios, de manera que dependiendo de si se ha podido crear un horario válido o no devuelve un Boolean diciendo que no se ha conseguido o un Boolean diciendo que se ha conseguido y el resultado.

- Limitaciones: Permite guardar i tener fácilmente accesibles las restricciones que afectan a todo el horario: la franja de trabajo y los días libres.
- Ocupaciones: Permite mantener constancia de las asignaciones que se van acumulando durante la generación del horario y todos los actores (niveles, asignaturas, grupos, subgrupos...) que intervienen en cada una de las horas de los días del horario.

### Algoritmo

El algoritmo empleado para generar los horarios es un algoritmo de backtracking que emplea forward checking para reducir en la mayor medida posible el nombre de posibles soluciones que se comprueba si son válidas basándose en 2 principios básicos:

- Se comprueba todas las restricciones lo antes posible, en cuanto la información necesaria para comprobar cada restricción esta definida para las asignaciones que se va a añadir a las posibles soluciones.
- Sólo se añade una asignación a una solución si esta es compatible con las asignaciones acumuladas en esa solución hasta el momento.

El algoritmo empieza generando todas las posibles Clases que se pueden crear en el escenario que se ha cargado en el programa. Una vez se generan las clases se ordenan de forma ascendente según el tamaño de la franja horaria en la que se pueden asignar a causa de las restricciones por las que se ven afectadas i este orden será el orden en el que las vayamos probando para generar posibles soluciones. (evitamos que asignaturas que son asignables a un rango mayor de horas no ocupen sin necesidad las franjas a las que se pueden asignar las que tienen un rango menor sin necesidad).

Una vez se elije una Clase para que forme parte de la Asignacion de una posible solución se prueba días para la Asignacion (descartando aquellos que son diaLibre o que causan que la Clase elegida en el momento incumpla una Restriccion), empezando por Domingo o Lunes alternativamente (dependiendo de si el número de Asignaciones definidas hasta el momento es par o impar).

Por cada Clase y día definidos se prueban las horas que no salen de la franja horaria que se ha calculado para la Clase en función de las Restricciones y que añadidas a las 2 variables anteriores no generan una Asignación que incumpla Restricciones.

Después, cuando ya tenemos Clase, día i hora para el candidato a Asignacion añadimos Aulas compatibles con la Sesión de la Clase involucrada y el número de plazas.

Este proceso se repite añadiendo Asignaciones hasta que nos quedamos sin Clases o hasta que no se es capaz de generar una Asignación válida. En el primer caso se devuelve la solución obtenida, en el 2o se retrocede 1 nivel y se prueba otra combinación de Clase, día, hora y Aula siguiendo el proceso descrito anteriormente.

## 4. Clases implementadas por miembro

### Enric Sosa

- Main
- CtrlDomain
- CtrlAsignaturasFile
- CtrlEscenariosDir
- CtrlPlanEstudiosFile
- CtrlRestriccionesFile
- CtrlPresentación
- CtrlPresentacionGenHorario
- VistaMantHorario
- driverAula
- driverGrupo
- driverPlanEstudios
- driverAsignatura
- driverClase
- driverNivel
- driverSesion
- driverSubGrupo
- driverAsignacion
- driverCorrequisito
- driverDiaLibre
- driverFranjaAsignatura
- driverFranjaNivel
- driverFranjaTrabajo
- driverNivelHora
- driverPrerrequisito
- driverSubGrupo
- driverDia
- driverHora

### Daniel Martín

- CtrlHorario
- LimitacionesHorario
- ReturnSet
- Asignacion
- Asignatura
- Aula
- Aux
- Clase
- Correquisito
- Dia
- DiaLibre
- FranjaAsignatura
- FranjaNivel
- FranjaTrabajo
- Grupo

- Hora
- Dia
- DiaLibre
- FranjaAsignatura
- FranjaNivel
- FranjaTrabajo
- Grupo
- Hora
- Horario
- Nivel
- NivelHora
- Ocupaciones
- PlanEstudios
- Prerrequisito
- Restriccion
- Sesion
- SubGrupo
- junitOcupaciones



## 5. Relación de librerías externas usadas

- Json-simple
- Hamcrest
- junit4

## 6. Manual

### Elegir función

Al iniciar el programa se muestran los comandos disponibles para el usuario:

1. Escoger escenario: Abre el menú de selección de escenarios donde se escoge que escenario usar para generar horarios.
2. Consultar horarios generados: Abre el menú de horarios donde se puede elegir que horario (de entre los que se han guardado previamente) mostrar por pantalla.
3. Salir: Finaliza la ejecución del programa.

Para elegir una de las 3 opciones solo se debe escribir el número de la opción elegida (1, 2 o 3) y seguidamente pulsar Enter.

### Escoger escenario

Al entrar en este menú se muestran todos los escenarios que ha detectado el programa y el comando “Atrás” para volver al menú anterior, asociados a un número. Debemos elegir un escenario o salir, para ello solo hay que escribir el número asociado a la opción que elijamos y pulsar Enter, después de mostrará el menú de generación de horario.

### Generar horario

En este menú se nos ofrecen las siguientes opciones:

1. Generar un documento .txt: Genera un horario con el escenario cargado, lo muestra por pantalla y lo guarda en un documento .txt. Luego vuelve al menú anterior.
2. Solamente mostrar el horario: Genera un horario con el escenario cargado y lo muestra por pantalla. Luego vuelve al menú anterior.
3. Atrás: Vuelve al menú anterior.

Para elegir una de las 3 opciones solo se debe escribir el número de la opción elegida (1, 2 o 3) y seguidamente pulsar Enter.

### Consultar horarios generados

En este menú se nos muestran todos los horarios guardados y el comando “Atrás” asociados a un número.

Para mostrar un horario o volver al menú anterior solo hay que escribir el número asociado a la opción que elijamos y pulsar Enter.

### Añadir escenario

Para añadir un escenario debemos crear una nueva carpeta con el nombre del escenario dentro de DATA.

Dentro de la carpeta se debe añadir 4 archivos json (asignaturas.json, aulas.json, pla\_estudis.json, restricciones.json) que han de contener la información del escenario deseada siguiendo el formato que podemos observar en los archivos con el mismo nombre del escenario Default (que no debe modificar-se).