

ECISeries

La **Escuela** quiere extender el sistema desarrollado para usar series en el proceso de formación de estudiantes. **ECISeries** ahora debe permitir (i) formar en aspectos adicionales a competencias e inteligencia emocional (como estaba anteriormente definido) , (ii) planificar la formación de estos aspectos por semestre y (iii) otorgar certificados especiales especiales asociados al logro de los diferentes aspectos (además de los créditos que se venían otorgando por tiempo). El sistema actualmente permite:

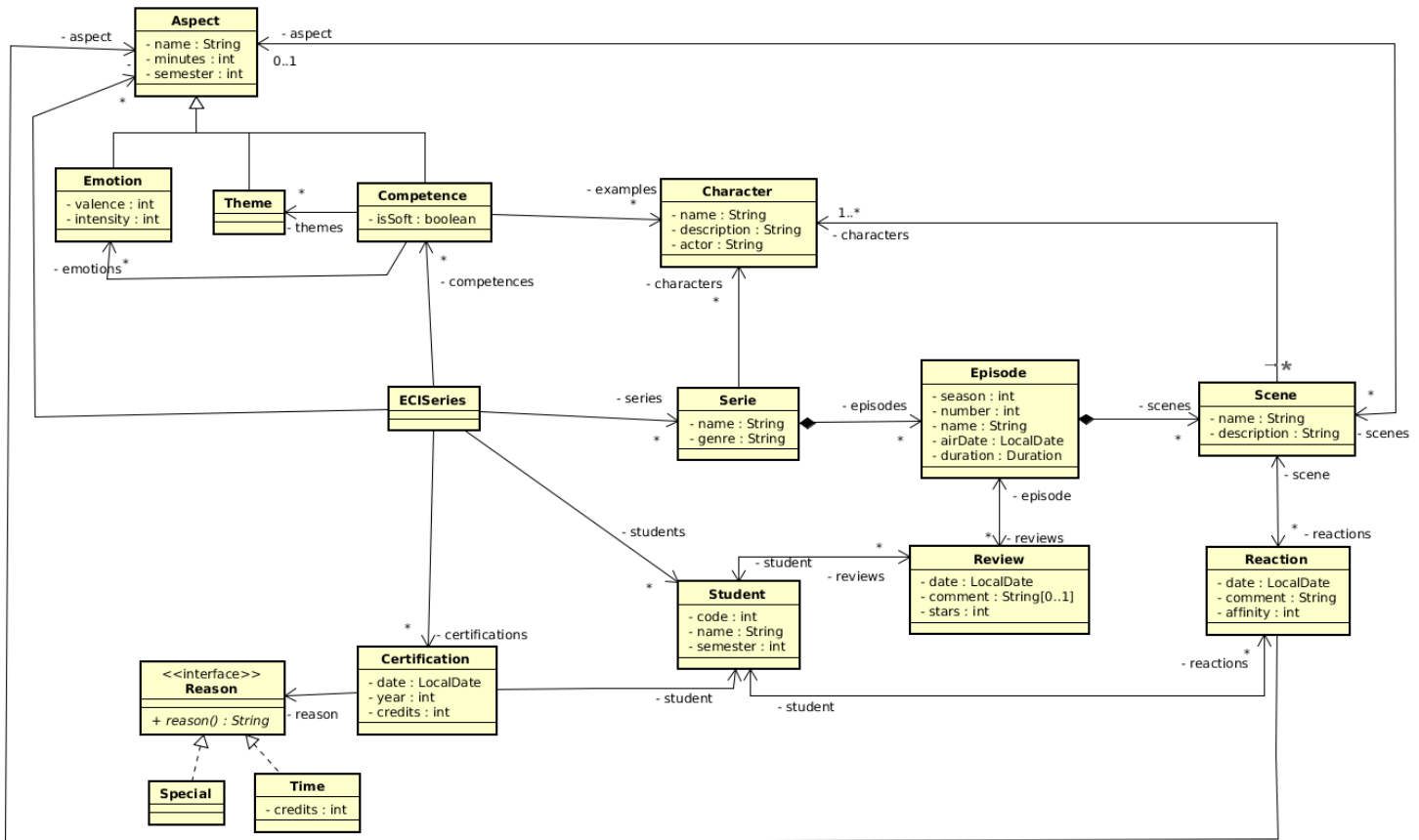
- Creación de series: La Escuela es la responsable de crear las diferentes series con toda su información.
- Registro de revisiones: Cuando un estudiante ve un episodio de alguna serie, puede registrar una revisión.
- Registro de reacciones: Cuando un estudiante siente que una escena ha tenido un impacto especial en él, puede registrar su reacción.
- Generar certificados de créditos. Cada año se deben generar los certificados de créditos a los estudiantes.

Las extensiones propuestas corresponden a dos de las funcionalidades.

- Registro de reacciones: Las reacciones de los estudiantes ahora serán hacia cualquiera de los aspectos (no sólo las emociones). En este momento se tienen tres aspectos : Emociones, Competencias y Conocimientos.
- Generar certificados: Ahora no sólo se realizarán certificados de créditos, sino certificados especiales sobre los diferentes aspectos. En el certificado se debe incluir el porque se otorga.

Para otorgar nuevos certificados se definieron los siguientes criterios:

- Las reacciones de los estudiantes se consideran válidas si reconocen el aspecto asociado a la misma con una afinidad mayor a 75.
- El certificado se otorga si el estudiante completa los minutos definidos para el mismo. El tiempo aproximado de cada escena es de 15 minutos.
- Los tiempos válidos de los estudiantes en cada aspecto se calcula de la siguiente manera:
 - Emociones. Se considera únicamente el tiempo de los episodios de la emoción con reacciones válidas.
 - Temas. Se calcula incluyendo el tiempo de los episodios con reacciones válidas y el de tiempo válido de todos sus sub-temas del semestre.
 - Competencias. Se calcula considerando el tiempo de los episodios de la competencia con reacciones válidas. Para que el tiempo sea reconocido debe haber invertido tiempo válidos en todos los temas y en al menos una de sus emociones con escenas definidas. Los temas y las emociones son del mismo semestre.



(Todos los contenedores son ArrayList)

En EciSeries

```
public boolean generateCertifications()
```

Genera las certificaciones de créditos del año actual

Solo se pueden generar las certificaciones en diciembre. Si ya se habian generado no se generan nuevamente.

Un crédito académico corresponde a 48 horas de trabajo,

Returns:

Si se generaron las certificaciones

I. (35%) PREPARANDO ASPECTOS

Implemente los métodos necesarios para cumplir con los compromisos asociados a la clase `Aspecto`.

MDD

- Estudie la especificación (documentación + encabezado) del método
- Realice el diagrama de secuencia (adicione el manejo de excepciones con otro color)
- Actualice el diagrama de clase con los nuevos elementos
- Escriba el código (adicione el manejo de excepciones con otro color)

En `aspecto`

```
public int goodReactions(Student s) throws ECISeriesExcepcion
    Calcula el número de reacciones del estudiante adecuadas para el aspecto
```

Returns:

Throws:

ECISeriesExcepcion- **EMPTY_SCENES** El aspecto no tiene las escenas necesarias definidas

NO_SEMESTER el estudiante no es del semestre correspondiente

```
public int minutes(Student s) throws ECISeriesException
    Retorna los minutos invertidos en el tema en el semestre correspondiente
```

Returns:

Throws:

ECISeriesExcepcion- **EMPTY_SCENES** El aspecto no tiene las escenas necesarias definidas

NO_SEMESTER el estudiante no es del semestre correspondiente

II. (20%) DISEÑANDO

Para generalizar la generación de certificados, diseñe el método

MDD

- Estudie la especificación (documentación + encabezado) del método
- Realice los diseños correspondientes (diagramas de secuencia, adicione el manejo de excepciones con otro color)
- Actualice el diagrama de clase con los nuevos elementos

En `EciSeries`

```
public boolean generateAllCertifications()
Genera todas las certificaciones del año actual
Solo se pueden generar las certificaciones en diciembre. Si ya se habían generado no se generan nuevamente.
```

Returns:

Si se generaron las certificaciones

Throws:

ECISeriesExcepcion- **EMPTY_SCENES** Si más del 50% de los aspectos no tiene las escenas necesarias definidas

III. (25%) EXTENDIENDO

Proponga la inclusión de un nuevo aspectos:

- Realice los cambios necesarios en el diagrama
- De los diseños resultantes del punto I y II, ¿Qué debe cambiar? ¿Qué no debe cambiar? ¿Qué debe extender?. Justifique su respuesta.
- Teniendo en cuenta los cambios necesarios y considerando el segundo principio SOLID. ¿Que fue bueno y que fue malo de su diseño?
- Realice los cambios y las extensiones definidas.

IV. (20%) CONCEPTOS

- ¿Qué diferencias existen entre una clase abstracta y una interfaz? ¿Cuándo usar cada una de ellas? (Detalle las diferencias considerando : atributos, métodos y objetos)
- ¿Cuáles son las diferentes relaciones entre componentes de un diagrama de clase? (Detalle nombre, significado y ejemplo)

Class <code>LocalDate</code> - Method Summary		Class <code>Duration</code> - Method Summary	
boolean	isAfter (<code>LocalDate</code> other) Checks if this date is after the specified date.	<code>Duration</code>	multipliedBy (<code>long</code> multiplicand) Returns a copy of this duration multiplied by the scalar.
boolean	isEqual (<code>LocalDate</code> other) Checks if this date is equal to the specified date.	<code>Duration</code>	plus (<code>Duration</code> duration) Returns a copy of this duration with the specified duration added.
Static <code>LocalDate</code>	now () Obtains the current date from the system clock	<code>Duration</code>	minus (<code>Duration</code> duration) Returns a copy of this duration with the specified duration subtracted.
int	getYear () Gets the year field.	Static <code>Duration</code>	ZERO Constant for a duration of zero.