

Les notes d'aquest examen i totals del curs (provisionals) es publicaran dimarts 15 de gener de 2019.

Le **revisions** es faran dijous 17 de gener d'11:30 a 13 al despatx Omega-133.

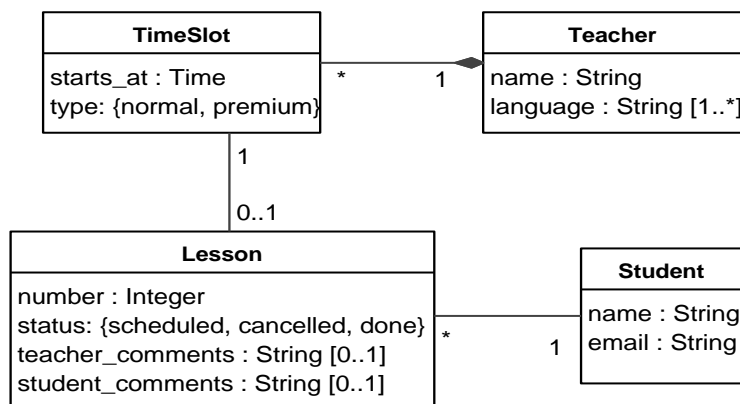
Pregunta 1 (1,5 punts): Justifiqueu per què l'API del vostre projecte està al nivell 2 del “*Richardson Maturity Model*”. Què caldria fer per arribar al nivell 3?

Pregunta 2 (1,5 punts): Responen **una sola** de les preguntes següents, i que correspongui a un tema **no presentat per vosaltres**:

- Explicueu els objectius principals de la monitorització d'aplicacions web.
- Explicueu les característiques més importants dels *Microframeworks*.

Exercici (7 punts)

Considereu un sistema per programar lliçons d'idiomes. Supposeu que els possibles clients (programes) dels serveis que dissenyareu ho faran des d'una intranet segura i que, per tant, no cal considerar el tema de l'autenticació/autorització. El diagrama de classes que en descriu el domini és el següent:



Identificadors: (Teacher, name), (TimeSlot, teacher.name+starts_at), (Lesson, number), (Student, email)

Caldrà dissenyar els serveis que permetin implementar les 5 operacions següents:

- Consultar la info (tots els atributs) d'un *Teacher* concret. Caldrà incloure també la info de tots els seus *TimeSlots* lliures. Un *TimeSlot* es considera lliure si no està associat a cap *Lesson* i comença com a mínim 24 hores més tard de l'hora actual del sistema.
- Consultar les *Lessons* que comencen dins d'un interval de temps

determinat ($start_time < end_time$) i obligatori. Hi ha d'haver també la possibilitat de filtrar-les per *Teacher*, *Student* i/o *status*. Per cada *Lesson*, cal retornar tots els seus atributs així com els de l'*Student* i *TimeSlot* corresponents, i també el *name* del *Teacher*.

- Eliminar un determinat *TimeSlot*. Cal que no estigui associat a cap *Lesson*.
- Programar una nova *Lesson*. Cal que el *TimeSlot* estigui lliure en aquell moment. El *number* el genera el sistema i l'*status* inicial és *scheduled*.
- Reprogramar una *Lesson*: s'associa un altre *TimeSlot* a la *Lesson*. Cal que aquest sigui un *TimeSlot* lliure del mateix *Teacher* i que la *Lesson* encara estigui en estat *scheduled*.

Més concretament,

- [2 punts]** Dissenyeu un servei web SOAP que inclogui les **4 primeres** operacions. Heu d'utilitzar els patrons RPC API i DTO. Per a cada operació cal definir els paràmetres d'entrada i el resultat. Utilitzeu el mode d'interacció *Request/Response*.
- [1 punt]** Dissenyeu un servei web SOAP utilitzant el mode d'interacció *Request/Acknowledge/Poll* per a l'operació de **reprogramar una Lesson**. Heu d'utilitzar els patrons RPC API i DTO.
- [3 punts]** Dissenyeu un servei web REST per a les **5 operacions**. Per descriure cada operació caldrà omplir la plantilla que teniu a continuació:

Title	Operation name	
URI	Example: /users or /users/:id	
Query Params	q=[String]: required opcional	
Method	GET POST DELETE PUT	
Body Data	(Only for POST and UPDATE requests) Example; {"name": String, "email": ArrayOf(String)}	
Returns	Success Response	Code and Content. Example: 200 OK { "id": String, "name": String, "email": ArrayOf(String) }
	Error Responses	Code and Content. Example: 404 Not Found { message: "no user exists with that id" }

- [1 punt]** Definiu l'escenari que utilitzi el servei web REST anterior i que permeti eliminar els *TimeSlots* lliures del *Teacher* amb *name* "Rance" que havien de començar durant el divendres 11 de gener de 2019.

a) Servei web SOAP per a les 4 primeres operacions

Operation name	Input	Output
getTeacherByName	getTeacherByNameReq= <name: String>	getTeacherByNameRes= <name: String, language: String [1..*], free_slots: TimeSlot [*]> where TimeSlot = <starts_at: Time, type: {normal, premium}>
getLessons	getLessonsReq= <interval: TimeInterval, teacher_name: String [0..1], student_email: String [0..1], status: {scheduled, ...} [0..1] > where TimeInterval = <start_time: Time, end_time: Time>	getLessonsRes= Set(InfoReservation) where InfoReservation= <number: Integer, status: {pending, ...} teacher_comments: String[0..1] student_comments: String[0..1] student: <name: String, email: String> slot: TimeSlot, teacher_name: String>
deleteTimeSlot	deleteTimeSlotReq= <teacher_name: String, starts_at: Time>	deleteTimeSlotRes= <code: Integer, message: String>
scheduleNewLesson	scheduleNewLessonReq= <teacher_name: String starts_at: Time student_email: String teacher_comments: String [0..1] student_comments: String [0..1]>	scheduleNewLessonRes= <code: Integer, message: String lesson_number: Integer [0..1]>

b) Servei web SOAP utilitzant el mode d'interacció Request/Acknowledge/Poll per a l'operació de reprogramar una Lesson:

Operation name	Input	Output
rescheduleLesson	rescheduleLessonReq= <num_lesson: Integer, starts_at: Time>	rescheduleLessonAck= <id: String>
getResults	rescheduleLessonAck = <id: String>	rescheduleLessonRes= <code: Integer, message: String>

c) Api REST per a les 5 operacions

Title	Consultar la info d'un Teacher concret	
URI	/api/teachers/:name	
Query Params		
Method	GET	
Body Data		
Returns	Success Response	200 OK <pre>{ "name:" "Perico Palotes" "language": ["english", "latin"] "free_slots": [{"starts_at": "2019-01-11T10:30:00.000Z", "type": "normal"}, ...] }</pre>
	Error Response	404 Not Found <pre>{message: "no Teacher with this name"}</pre>

Title	Consultar lessons	
URI	/api/lessons	
Query Params	start_time=[String]: required end_time=[String]: required teacher_name=[String]: optional student_email=[...]: optional status=[...]: optional	
Method	GET	
Body Data		
Returns	Success Response	200 OK <pre>[{ "number": 667766, "status": "scheduled", "teacher_comments": "Bring your notebook", "student_comments": null, "student": { "name": "Manolo", "email": "manolo@g.cc"}, "slot": { "starts_at": "2019-01-10T11:30:00.000Z", "type": "normal"}, "teacher_name": "Rance" }, ...]</pre>
	Error Response	400 Bad Request <pre>{message: "incorrect interval"}</pre>

Title	Eliminar un slot concret	
URI	/api/teachers/:name/time_slots/:starts_at	
Query Params		
Method	DELETE	
Body Data		
Returns	Success Response	204 No content
	Error Response	404 Not Found {message: "no time slot for this teacher and start time"} 409 Conflict {message: "time slot assigned to a Lesson"}

Title	Programar una nova lesson	
URI	/api/lessons	
Query Params		
Method	POST	
Body Data	{ "teacher_name": "Rance", "starts_at": "2019-01-10T11:30:00.000Z", "student_email": "manolo@g.cc", "teacher_comments": "Bring your notebook" }	
Returns	Success Response	201 Created { "number": 667766, "status": "scheduled", "teacher_comments": "Bring your notebook", "student_comments": null, "student": { "name": "Manolo", "email": "manolo@g.cc" }, "slot": { "starts_at": "2019-01-10T11:30:00.000Z", "type": "normal" }, "teacher_name": "Rance" }
	Error Response	400 Bad request {message: "wrong parameters"} 409 Conflict {message: "Time slot is not free", etc.}

Title	Reprogramar una lesson	
URI	/api/lessons/:number	
Query Params		
Method	PUT	
Body Data	{ "starts_at": "2019-01-14T10:30:00.000Z" }	
Returns	Success Response	200 OK { "number": 667766, "status": "scheduled", "teacher_comments": "Bring your notebook", "student_comments": null, "student": { "name": "Manolo", "email": "manolo@g.cc" }, "slot": { "starts_at": "2019-01-14T10:30:00.000Z", "type": "premium" }, "teacher_name": "Rance" }
	Error Response	404 Not Found {message: "no lesson for this number"} 400 Bad request {message: "invalid time format"} 409 Conflict {message: "the new time slot is not free", etc.}

d) Client

1. Obtinc tota info (slots lliures inclosos) del teacher amb name = "Rance":

GET /api/teachers/**Rance**

2. Per cada item de l'array free_slots tal que el seu atribut starts_at = **TS** comença per "2019-01-11" faig la petició següent:

DELETE /api/teachers/**Rance**/time_slots/**TS**