

Les notes d'aquest examen i totals del curs (provisionals) es publicaran dilluns 22 de gener de 2018.

La revisió es farà dimecres 24 de gener de 15 a 16 al despatx Omega-133.

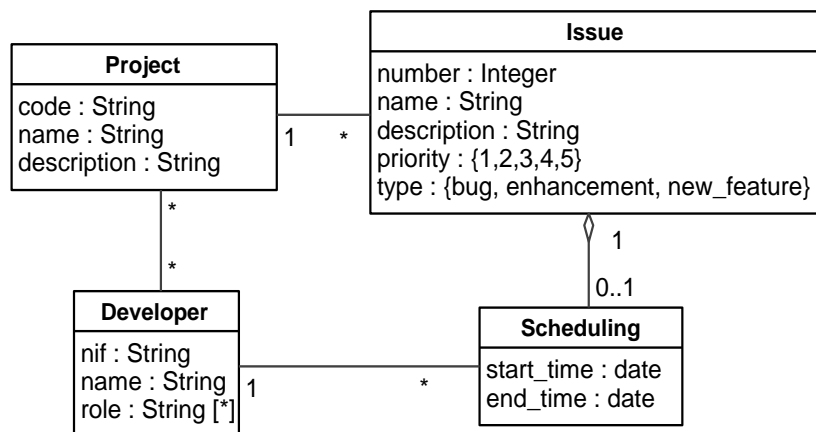
**Pregunta 1 (1,5 punts):** Expliqueu per què l'API de Twitter està al nivell 1 del "Richardson Maturity Model". Com hauria de ser per estar al nivell 3?

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

- Expliqueu les diferències entre els 2 tipus de monitoritzacions: sintètica i passiva.
- Expliqueu els beneficis de les arquitectures basades en *microservices*.

### Exercici (7 punts)

Considereu un sistema per planificar (*schedule*) *issues*. 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: (Project, code), (Issue, project.code+number), (Scheduling, issue.project.code+issue.number), (Developer, nif)

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

- Consultar la info (tots els atributs) d'un *issue* concret. Si l'*issue* ja està planificat (té un *scheduling*), caldrà incloure-hi també els atributs *start\_time*, *end\_time*, així com els *nif* i *name* del *developer*.
- Consultar els *issues* d'un determinat *project* amb la possibilitat de filtrar-los

per *type*, *priority* i/o *developer* (nif). Per cada *issue*, cal retornar la mateixa informació que a l'operació anterior.

- Des-planificar un *issue* (és a dir, eliminar el seu *scheduling*).
- Modificar *start\_time* i/o *end\_time* d'un *scheduling*. Cal que *start\_time* ≤ *end\_time*.
- Planificar un *issue* (és a dir, crear un *scheduling*), indicant tota la info necessària. Cal que l'*issue* no estigui ja planificat, que *start\_time* ≤ *end\_time* i que el *developer* pertanyi al *project*.

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 planificar un *issue*. 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 tots els *schedulings* que el *developer* "66234111X" (nif) té assignats en el *project* "2016LLAVOR009" (code).

a) Servei web SOAP per a les 4 primeres operacions

Operation name	Input	Output	Description
getIssueById	getIssueByIdReq= <id: IssueID>  where  IssueID = <prj_code: String, iss_num: Integer>	getIssueByIdRes= <id: IssueID, name: String, description: String, priority: {...}, type: {...}>, ..scheduling: <start_time: ..., end_time: ..., dev_nif: ..., dev_name: ...>[0..1]>	Get the complete info of an issue given its id
getIssues	getIssuesReq= <prj_code: String type: {...}[0..1], priority: {...}[0..1], dev_nif: String [0..1]>	getIssuesRes= Set(getIssueByIdRes)	Get the complete info of the issues filtered by the provided params (if any)
unSchedule	unScheduleReq= <issue_id: IssueID>	unScheduleRes= <code: Integer, message: String>	Unschedule an issue
modifyScheduling	modifySchedulingReq= <issue_id: IssueID, start_time: Date[0..1], end_time: Date[0..1]>	modifySchedulingRes= <code: Integer, message: String>	Modify some attributes of a given Scheduling

b) Servei web SOAP utilitzant el mode d'interacció Request/Acknowledge/Poll per a l'operació de planificar un issue

Operation name	Input	Output	Description
schedule	scheduleReq= <issue_id: IssueID, dev_nif: String, ..start_time: Date, end_time: Date>	scheduleAck= <id: String>	Request to create a new Scheduling for a given issue.
getResults	scheduleAck= <id: String>	scheduleRes= <code: Integer, message: String>	S'obté el resultat de la petició anterior

c) Api REST per a les 5 operacions

Title	Consultar la info d'un issue concret	
URI	/api/projects/:code/issues/:number	
Query Params		
Method	GET	
Body Data		
Returns	Success Response	200 OK <pre>{ "proj_code": ":"SUPERSEDE"   "number": "9780",   "name": "Login bug",   "description": "When a ...",   "priority": 4, "type": "bug",   .. "scheduling": { "dev_nif": "98988945K",                     "dev_name": "Manolo R",                     "start_time": "2017-04-01",                     "end_time": "2017-04-15"} }</pre>
	Error Response	404 Not Found <pre>{message: "no issue with these code &amp; number"}</pre>

Title	Consultar issues	
URI	/api/projects/:code/issues	
Query Params	priority=[...]: optional type=[...]: optional dev_nif=[String]: optional	
Method	GET	
Body Data		
Returns	Success Response	200 OK <pre>[ /* array of issue responses */ ]</pre>
	Error Response	400 Bad Request <pre>{message: "incorrect type/priority. "}</pre> 404 Not Found <pre>{message: "no project with this code"}</pre>

Title	Desplanificar un issue	
URI	/api/projects/:code/issues/:number/scheduling	
Query Params		
Method	DELETE	
Body Data		
Returns	Success Response	204 No content
	Error Response	404 Not Found <pre>{message: "no issue with that code &amp; number"}</pre>

Title	Canviar un scheduling	
URI	/api/projects/:code/issues/:number/scheduling	
Query Params		
Method	PUT	
Body Data	{ "start_date": 2018-01-15, "end_date": 2018-02-23 }	
Returns	Success Response	200 OK { "start_date": 2018-01-15, "end_date": 2018-02-23 }
	Error Response	404 Not Found {message: "no schedulling for these code & number"} 400 Bad request {message: "wrong dates"}

Title	Planificar un issue	
URI	/api/projects/:code/issues/:number/scheduling	
Query Params		
Method	POST	
Body Data	{ "dev_nif": "98988945K", "start_time": "2017-04-01", "end_time": "2017-04-15" }	
Returns	Success Response	201 Created
	Error Response	404 Not Found {message: "no issue with these code & number"} 400 Bad request {message: "wrong dates / developer does not work for project, / missing parameters"} 409 Conflict {message: "the issue was already schedulled"}

#### d) Client

1. Obtinc tots els issues del project "2016LLAVOR009" estan planificades per ser implementades per al developer "66234111X"

GET /api/projects/**2016LLAVOR009**/issues?dev\_nif=**66234111X**

2. Per cada issue amb number = XX obtingut, faig la petició següent:

DELETE /api/projects/**2016LLAVOR009**/issues/**XX**/scheduling