
PROGETTO SIGEOL

Piano di Qualifica

(Delta)

Redazione: Carlo Scortegagna, Barbiero Mattia

9 giugno 2009



quixoft.sol@gmail.com

Verifica:	Scarpa Davide
Approvazione:	Beggiato Andrea
Stato:	Formale
Uso:	Esterno
Distribuzione:	QuiXoft
	Rossi Francesca
	Vardanega Tullio
	Conte Renato

Sommario

Aggiornamento del Piano di Qualifica per il progetto "SIGEOL",
contenente solamente le modifiche rispetto al documento consegnato alla
Revisione di Qualifica.



Indice

1	Introduzione	1
1.1	Scopo del documento	1
2	Resoconto delle attività di verifica	1
2.1	Tracciamento componenti - requisiti	1
2.2	Dettaglio delle verifiche tramite test	1
2.2.1	Unit Tests	1
2.2.2	Functional Tests	18
2.3	Metriche	30
2.3.1	Rcov	30
2.3.2	Flay	30
2.3.3	Flog	31
2.3.4	Saikuro	32
2.3.5	Reek	32
2.3.6	Roodi	33
2.3.7	Churn	33



1 Introduzione

1.1 Scopo del documento

Lo scopo del presente documento è di notificare gli aggiornamenti che sono stati apportati rispetto al documento consegnato alla precedente revisione.

Tali modifiche rappresentano il proseguimento delle attività di verifica e validazione del materiale prodotto e la descrizione degli ambienti di prova e di collaudo.

2 Resoconto delle attività di verifica

Ogni risultato ottenuto dalle attività di verifica, validazione e qualifica dovrà essere attentamente elencato in questa sezione, in modo da assicurare che tutti i problemi e le relative soluzioni siano tracciate per garantire la massima qualità possibile del prodotto finale.

2.1 Tracciamento componenti - requisiti

2.2 Dettaglio delle verifiche tramite test

Per l'esecuzione delle attività di verifica tramite test verranno utilizzati i moduli e le classi rese disponibili dal framework Rails, e più precisamente ogni test di unità dovrà estendere la classe ActiveSupport::TestCase ed ogni test funzionale dovrà estendere la classe ActionController::TestCase.

2.2.1 Unit Tests

Nello sviluppo di un'applicazione tramite il framework Rails, i test di unità (unit test) sono specifici per la verifica degli elementi appartenenti al componente Model. Data la particolare importanza di questa componente in quanto gestisce la persistenza dei dati, il team QuiXoft ha scelto di effettuare gli unit tests attraverso l'uso di fixtures, ovvero con istanze reali di un determinato model opportunamente salvate in file con estensione `yml` all'interno della directory `test/fixtures`. Prima dell'esecuzione della verifica il database di test sarà inizializzato con i dati contenuti nelle fixtures che vengono utilizzate in quel particolare test.

Ogni classe che implementa un insieme di test per un particolare model dovrà essere denominata `NomeModelTest` ed essere salvata su di un file chiamato `nome_model_test.rb` all'interno della directory `test/unit`.

Un esempio di unit test per il model `address` e delle relative fixtures utilizzate è dato dalla seguente porzione di codice:

nel file `address_test.rb`

```
class AddressTest < ActiveSupport::TestCase
```



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

```
fixtures :addresses, :buildings

def setup
  @a=Address.new
end

def test_correct_address
  @a.telephone="049-9050231"
  @a.street="Via Belzoni 6"
  @a.city="Villafranca veronese"
  assert @a.save
end

def test_destroy_address_in_building_address_id_must_nil
  addresses(:address_2).delete
  !assert(buildings(:building_1).address_id)
end
end

  nel file addresses.yml

address_2:
  city: Vicenza
  telephone: 049-9075393
  street: via rossini 9

  nel file buildings.yml

building_1:
  name: Torre Archimede
  address: address_2
```

Di seguito sono riportate le tabelle che riassumono l'intera campagna di test delle classi appartenenti al componente model.

Test su AcademicOrganization



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
1.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
2.1	Il contenuto dell'attributo number dell'oggetto d'istanza è 0.	number può contenere solo valori interi compresi tra 1 e 4. Non essendo valido l'oggetto non dovrà essere salvato nel database.	✓
2.2	number contiene il valore 7.	Lo stesso del caso di prova 2.1.	✓
2.3	number contiene il valore 3.	number contiene un valore corretto e quindi non devono essere riscontrati errori di validazione su di esso.	✓
3.1	Il contenuto dell'attributo name dell'oggetto d'istanza è 12a34.	name può contenere solo caratteri alfabetici. Non essendo valido l'oggetto non dovrà essere salvato nel database.	✓
3.2	name contiene un valore valido.	Non devono essere riscontrati errori di validazione su name.	✓
4.1	Agli attributi name e number dell'oggetto da salvare nel db vengono assegnati gli stessi valori di una tupla già presente.	Non devono esistere più tuple con la stessa coppia di valori contenuti in name e number. Non essendo valido l'oggetto non dovrà essere salvato nel database.	✓

Test su Address



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
5.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
6.1	Il contenuto dell'attributo telephone è 12345-123456.	Il prefisso ha più di quattro cifre e quindi non è valido. L'oggetto per questo motivo non deve essere salvato nel db.	✓
6.2	telephone contiene il valore 1234-12345.	Il prefisso è corretto ma il numero contiene meno di sei cifre. L'oggetto per questo motivo non deve essere salvato nel db.	✓
6.3	telephone contiene il valore 1a23-12345.	Contiene un carattere alfabetico. L'oggetto per questo motivo non deve essere salvato nel db.	✓
6.4	telephone rispetta l'espressione regolare.	Non devono essere riscontrati errori di validazione su telephone.	✓
7.1	Cancellazione di un indirizzo associato ad uno user.	Eliminato l'indirizzo, lo user associato deve avere il contenuto della chiave esterna(address_id) a nil.	✓
8.1	Cancellazione di un indirizzo associato ad un palazzo.	Eliminato l'indirizzo, il palazzo associato deve avere il contenuto della chiave esterna(address_id) a nil.	✓
9.1	Creazione e salvataggio di un indirizzo valido.	L'oggetto deve essere salvato nel database.	✓

2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su BooleanConstraint

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
10.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
11.1	L'attributo bool dell'oggetto d'istanza contiene il valore 3.	Il valore 3 deve essere convertito a false.	✓
11.2	bool assume il valore 1	Il valore 1 deve essere convertito a true.	✓
11.3	bool assume il valore 0.	Il valore 0 deve essere convertito a false.	✓
12.1	L'attributo isHard dell'oggetto d'istanza contiene il valore 11.	isHard deve contenere solo valori interi compresi tra 0 e 10. L'oggetto per questo motivo non deve essere salvato nel db.	✓
12.2	isHard contiene il valore negativo -1.	Lo stesso del caso di prova 12.1	✓
12.3	isHard contiene il valore 1.1.	Lo stesso del caso di prova 12.1	✓
12.4	isHard contiene il valore 0.	Non devono essere riscontrati errori di validazione su isHard.	✓

Test su Building

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
13.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
14.1	L'attributo name dell'oggetto da salvare assume un valore uguale a quello di un oggetto già salvato.	Non devono esistere due palazzi con lo stesso nome. Quindi l'istanza del caso di prova non essendo valida non deve essere salvata nel db.	✓
15.1	Eliminazione di un oggetto di tipo Building.	Non deve più esistere nel database il palazzo, l'indirizzo associato e tutte le classi appartenenti.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su Capability

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
16.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
17.1	L'attributo name dell'oggetto da salvare assume un valore uguale a quello di un oggetto già salvato.	Una capability non può avere lo stesso nome di un'altra. Quindi l'istanza del caso di prova non essendo valida non deve essere salvata nel db.	✓

Test su Classroom

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
18.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
19.1	L'attributo name dell'oggetto da salvare assume un valore uguale a quello di un oggetto già salvato. Inoltre entrambi possiedono lo stesso valore di building_id(chiave esterna per building)	In un stesso palazzo non possono esistere classi con lo stesso nome. L'istanza del caso di prova non essendo valida non deve essere salvata nel db.	✓
20.1	Eliminazione di un oggetto di tipo Classroom.	Non devono più esistere nel database la classe e tutti i vincoli associati.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su Curriculum

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
21.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
22.1	L'attributo name dell'oggetto da salvare assume un valore uguale a quello di un oggetto già salvato. Inoltre entrambi possiedono lo stesso valore di graduate_course_id(chiave esterna per graduate_course)	In un stesso corso di laurea non possono esistere due curricula con lo stesso nome. L'istanza del caso di prova non essendo valida non deve essere salvata nel db.	✓
23.1	Un insegnamento(oggetto di tipo Teacher) è associato a due curricula. Cancellazione di uno dei due.	Dopo l'operazione di eliminazione, l'insegnamento deve essere ancora presente nel db.	✓
23.2	Cancellazione dell'altro curriculum.	Dopo l'operazione di eliminazione l'insegnamento non deve essere più presente nel db.	✓
24.1	Eliminazione di un oggetto di tipo Curriculum associato ad un insegnamento.	Non devono più esistere nel database il curriculum e l'insegnamento associato dato che non apparteneva ad altri curricula.	✓

Test su DidacticOffice

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
25.1	Eliminazione di un oggetto di tipo DidacticOffice(Segreteria didattica).	Non devono più esistere nel database la segreteria e lo user associato.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su ExpiryDate

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
26.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
27.1	L'attributo date dell'oggetto d'istanza contiene una data precedente alla data di sistema.	Per essere valida la data deve essere maggiore o uguale alla data di sistema. L'oggetto per questo motivo non deve essere salvato nel db.	✓
28.1	L'attributo period dell'oggetto d'istanza contiene il valore decimale 1.1	period deve contenere un intero compreso tra uno e quattro. L'oggetto per questo motivo non deve essere salvato nel db.	✓
28.2	period contiene il valore negativo -1.	Lo stesso del caso di prova 28.1.	✓
28.3	period contiene il valore 11.	Lo stesso del caso di prova 28.1.	✓
28.4	periodo contiene il valore 6.	Non devono essere riscontrati errori di validazione su period.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su GraduateCourse

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
29.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
30.1	L'attributo duration dell'oggetto d'istanza contiene un valore negativo pari a -1.	Per essere valido duration deve contenere un intero compreso tra uno e sei. L'oggetto per questo motivo non deve essere salvato nel db.	✓
30.2	duration contiene il valore decimale 1.1.	Lo stesso del caso di prova 30.1.	✓
30.3	duration contiene il valore 7.	Lo stesso del caso di prova 30.1.	✓
30.4	duration contiene il valore 6.	Non devono essere riscontrati errori di validazione su duration.	✓
31.1	Eliminazione di un corso di laurea(GraduateCourse) associato a due curricula. Entrambi sono associati ad uno stesso insegnamento(Teaching).	Non devono più esistere nel database il corso di laurea, i due curricula e l'insegnamento associato.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su Period

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
32.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
33.1	L'attributo year dell'oggetto d'istanza contiene un valore negativo pari a -1.	Per essere valido year deve contenere un intero compreso tra uno e sei. L'oggetto per questo motivo non deve essere salvato nel db.	✓
33.2	year contiene il valore decimale 1.2.	Lo stesso del caso di prova 33.1.	✓
33.3	year contiene il valore 7.	Lo stesso del caso di prova 33.1.	✓
33.4	year contiene il valore 6.	Non devono essere riscontrati errori di validazione su year.	✓
34.1	L'attributo subperiod dell'oggetto d'istanza contiene un valore negativo pari a -1.	Per essere valido subperiod deve contenere un intero compreso tra uno e quattro. L'oggetto per questo motivo non deve essere salvato nel db.	✓
34.2	subperiod contiene il valore decimale 1.2.	Lo stesso del caso di prova 34.1.	✓
34.3	subperiod contiene il valore 5.	Lo stesso del caso di prova 34.1.	✓
34.4	subperiod contiene il valore 4.	Non devono essere riscontrati errori di validazione su subperiod.	✓
35.1	A subperiod ed year vengono assegnati gli stessi valori di un oggetto già salvato.	Non possono esistere due periodi con gli stessi valori di subperiod ed year. L'istanza del caso di prova non essendo valida non deve essere salvata nel db.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su QuantityConstraint

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
36.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
37.1	L'attributo isHard dell'oggetto d'istanza contiene il valore 11.	isHard deve contenere solo valori interi compresi tra 0 e 10. L'oggetto per questo motivo non deve essere salvato nel db.	✓
37.2	isHard contiene il valore -1.	Lo stesso del caso di prova 12.1	✓
37.3	isHard contiene il valore 1.1.	Lo stesso del caso di prova 12.1	✓
37.4	isHard contiene il valore 0.	Non devono essere riscontrati errori di validazione su isHard.	✓
38.1	L'attributo quantity dell'oggetto d'istanza contiene il valore 1001.	quantity deve contenere solo valori interi compresi tra 1 e 1000. L'oggetto per questo motivo non deve essere salvato nel db.	✓
38.2	quantity contiene il valore -1.	Lo stesso del caso di prova 38.1	✓
38.3	quantity contiene il valore 1.1.	Lo stesso del caso di prova 38.1	✓
38.4	quantity contiene il valore 1.	Non devono essere riscontrati errori di validazione su quantity.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su TeacherMailer

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
39.1	Creazione di una mail contenente le istruzioni per l'attivazione di uno User tramite il metodo activate_teacher(sender, receiver). Sender e Receiver sono due oggetti di tipo User opportunamente creati che corrispondono al mittente e al destinatario.	L'indirizzo del destinatario della mail deve essere uguale al contenuto dell'attributo mail di receiver.	✓
39.2	Stessa descrizione per il caso di prova 39.1.	L'indirizzo del mittente della mail deve essere uguale al contenuto dell'attributo mail di sender.	✓
39.3	Stessa descrizione per il caso di prova 39.1.	Il subject della mail deve essere il seguente: Creazione account SIGEOL.	✓
39.4	Stessa descrizione per il caso di prova 39.1.	L'url necessario all'attivazione dell'account, presente nel corpo della mail, deve essere originato correttamente.	✓

Test su Teacher

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
40.1	Aggiornamento di un oggetto di tipo Teacher lasciando degli attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido e quindi non aggiornare la corrispondente tupla nel database.	✓
41.1	Eliminazione di un oggetto Teacher.	Non devono più esistere nel database il docente, lo user associato e tutti i vincoli e le preferenze.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su Teaching

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
42.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
43.1	cfu, labhours, classhours e studentsNumber contengono il valore 12.5.	Tutti e quattro devono contenere un valore numerico intero positivo. L'oggetto non è valido e non deve essere salvato nel db.	✓
44.1	cfu, labhours, classhours e studentsNumber contengono il valore -12.	Lo stesso del caso di prova 43.1.	✓
45.1	cfu assume il valore 21, labhours e classhours il valore 51 e studentsNumber il valore 1001.	Tutti e quattro gli attributi superano di un'unità le soglie imposte e quindi deve essere riscontrato un errore di validazione su ognuno di essi.	✓
46.1	Creazione di un insegnamento associato ad un periodo non compatibile con l'organizzazione accademica del corso di laurea dell'insegnamento.	L'oggetto d'istanza non essendo valido non deve essere salvato nel database.	✓

Test su TemporalConstraint

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
47.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
48.1	L'attributo day dell'oggetto d'istanza contiene il valore 6.	day deve contenere solo valori interi compresi tra 1 e 5. L'oggetto per questo motivo non deve essere salvato nel db.	✓
48.2	day contiene il valore 0.	Lo stesso del caso di prova 48.1.	✓
48.3	day contiene il valore 1.5.	Lo stesso del caso di prova 48.1.	✓
48.4	day contiene il valore 1.	Non deve essere riscontrato nessun errore di validazione su day.	✓
49.1	L'attributo isHard dell'oggetto d'istanza contiene il valore 11.	isHard deve contenere solo valori interi compresi tra 0 e 10. L'oggetto per questo motivo non deve essere salvato nel db.	✓
49.2	isHard contiene il valore -1.	Lo stesso del caso di prova 49.1	✓
49.3	isHard contiene il valore 1.1.	Lo stesso del caso di prova 49.1	✓
49.4	isHard contiene il valore 0.	Non devono essere riscontrati errori di validazione su isHard.	✓
50.1	Il valore contenuto in startHour assume un valore più grande di quello in endHour	Ovviamente l'ora d'inizio(startHour) dell'indisponibilità non può essere più grande dell'ora fine(endHour). Quindi non essendo valido l'oggetto non deve essere salvato nel db.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su TimetableEntry

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
51.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
52.1	L'attributo day dell'oggetto d'istanza contiene il valore 6.	day deve contenere solo valori interi compresi tra 1 e 5. L'oggetto per questo motivo non deve essere salvato nel db.	✓
52.2	day contiene il valore 0.	Lo stesso del caso di prova 52.1.	✓
52.3	day contiene il valore 1.5.	Lo stesso del caso di prova 52.1.	✓
52.4	day contiene il valore 1.	Non deve essere riscontrato nessun errore di validazione su day.	✓
53.1	Il valore contenuto in startTime assume un valore più grande di quello in endTime	Ovviamente l'ora d'inizio(startTime) di un riga dello schema d'orario non può essere più grande dell'ora fine(endHour). Quindi non essendo valido l'oggetto non deve essere salvato nel db.	✓
54.1	Creazione di un oggetto TimetableEntry con gli stessi valori di day, StartTime, EndTime, classroom_id e timetable_id di un oggetto già salvato.	Per evitare sovrapposizioni, in uno stesso schema d'orario identificato da timetable_id non possono esistere righe con ugual giorno, ora di inizio, ora di fine e classe occupata. L'istanza appena creata non essendo valida non deve essere salvata nel db.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su Timetable

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
55.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
56.1	L'attributo year dell'oggetto d'istanza contiene il valore 20000-20.	il contenuto di year non è valido. L'oggetto per questo motivo non deve essere salvato nel db.	✓
56.2	year contiene 2007-08.	year, per essere valido deve assumere nei primi quattro caratteri(2007) un valore successivo all'anno di sistema-1. L'anno di sistema nel caso di prova corrisponde a 2009. Non essendo valido, l'oggetto non deve essere salvato nel db	✓
56.3	year contiene 2008-10.	year per essere valido doveva avere nelle ultime due cifre i caratteri 09. Non essendo valido, l'oggetto non deve essere salvato nel db	✓
56.4	year contiene 2008-09.	Non deve essere riscontrato nessun errore di validazione su year.	✓
57.1	Creazione di un oggetto Timetable con gli stessi valori di year, graduate_course_id(chiave esterna per il corso di laurea) e period_id(chiave esterna per il periodo) di un oggetto già salvato.	Non possono esistere per uno stesso corso di laurea, periodo e anno più schemi d'orario.	✓

2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su User

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
58.1	Istanza di un oggetto con attributi nulli.	Il sistema deve riconoscere l'oggetto come non valido.	✓
59.1	L'attributo password dell'oggetto d'istanza è vuoto.	password deve essere di almeno sei caratteri di tipo alfanumerico più il carattere'. '. L'oggetto per questo motivo non deve essere salvato nel db.	✓
59.2	password contiene la stringa prova.	prova contiene meno di sei caratteri. Non essendo valido, l'oggetto non deve essere salvato nel db	✓
59.3	password contiene pro.va.	Non deve essere riscontrato nessun errore di validazione su password.	✓
60.1	L'attributo mail dell'oggetto d'istanza contiene la stringa prova?id=1@math.unipd.it	mail contiene al suo interno un carattere non valido:?. Non essendo valido, l'oggetto non deve essere salvato nel db.	✓
60.2	L'attributo mail contiene la stringa agrossel@math.unipd.it.	Non deve essere riscontrato nessun errore di validazione su mail.	✓
61.1	Creazione di uno user con lo stesso valore dell'attributo mail di un altro oggetto già salvato.	Non possono esistere due user con la stessa mail; quindi l'oggetto appena istanziato non è valido e non deve essere salvato.	✓
62.1	Creazione di uno user con password uguale ad alessandro.	L'oggetto istanziato viene correttamente salvato nel database. Il contenuto di password deve essere uguale alla stringa alessandro crittografata con algoritmo SHA1.	✓
63.1	Creazione di uno user senza specificarlo.	Un oggetto di tipo user deve appartenere o ad una segreteria o ad un insegnante; se non è associato a nulla l'istanza non è valida e non deve essere salvata nel db.	✓

2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
64.1	Al metodo <code>authenticate</code> vengono passati uno user ed una password non corretti.	Il metodo deve ritornare un valore booleano pari a <code>false</code> .	✓
64.2	Ora vengono passati uno user ed una password corretti.	Il metodo deve ritornare un valore booleano pari a <code>true</code> .	✓
65.1	Eliminazione di un oggetto User.	Non devono più esistere nel database lo user, l'indirizzo e il docente o la segreteria associata.	✓
66.1	Creazione di uno user associato a tutte le possibili capabilities(privilegi).	I metodi che iniziano con <code>manage_</code> di un oggetto User ritornano <code>true</code> se l'istanza possiede quel determinato privilegio. In questo caso di prova ogni metodo deve ritornare <code>true</code> .	✓
67.1	Creazione di uno user associato ad una segreteria didattica.	Lo user appartiene ad una segreteria quindi il metodo <code>own_by_didactic_office?</code> deve ritornare <code>true</code> e <code>own_by_teacher?</code> <code>false</code> .	✓

Test su Belong

ID CASO DI PROVA	DESCRIZIONE	OBIETTIVO	ESITO
69.1	Creazione di un oggetto di tipo Belong che associa un curriculum ad un insegnamento già associati.	Esistendo già l'associazione, l'istanza non deve essere salvata.	✓

2.2.2 Functional Tests

Nello sviluppo di un'applicazione tramite il framework Rails, i test funzionali (functional tests) sono specifici per la verifica degli elementi appartenenti alla componente Controller. Dato che gli unit tests sono stati effettuati tramite istanze reali, il team QuiXoft ha scelto di utilizzare la strategia dei Mock objects per la verifica delle azioni presenti nei controller.

Ogni classe che implementa un insieme di test per un particolare controller dovrà essere denominata `NomeControllerTest` ed essere salvata su di un file chiamato `nome_controller_controller_test.rb` all'interno della directory `test/functional`.



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Un esempio di functional test per il controller `sessions` è dato dalla seguente porzione di codice:

nel file `sessions_controller_test.rb`

```
class SessionsControllerTest < ActionController::TestCase

  test "Guest usa New" do
    get :new
    assert_template "new"
    assert_response :success
  end

  test "Immissione di email e password validi" do
    user = stub(:id => :an_id, :mail => "a_mail",
               :password => "a_password")
    User.stubs(:authenticate).returns(user)
    user.stubs(:active?).returns(true)
    post :create, :mail => user.mail, :password => user.password
    assert_equal session[:user_id], :an_id
    assert_redirected_to timetables_url
  end
end
```

Di seguito sono riportate le tabelle che riassumono la campagna di test delle classi appartenenti al componente controller. Dato il gran numero di test effettuati(186 tests) si elencheranno solo i più significativi.



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su BuildingsController

ID TEST	DESCRIZIONE	OBIETTIVO	ESITO
da 1 a 5	Un utente non autenticato(guest) richiede l'esecuzione di azioni riservate ad utenti con privilegi.	L'utente dev'essere indirizzato alla pagina di login.	✓
7	Un utente autenticato con privilegio di modificare edifici richiede l'esecuzione dell'azione new .	L'utente dev'essere indirizzato alla pagina di creazione edificio.	✓
8	Lo stesso tipo di utente del test 7 richiede l'esecuzione dell'azione edit .	L'utente dev'essere indirizzato alla pagina di modifica dei dati relativi ad un edificio.	✓
9	Lo stesso tipo di utente del test 7 crea correttamente un edificio.	L'utente dev'essere indirizzato alla pagina di amministrazione edifici.	✓
da 10 a 11	Lo stesso tipo di utente del test 7 tenta di creare un edificio con attributi non validi.	L'utente non dev'essere indirizzato, deve rimanere nella pagina di creazione edificio.	✓
12	Lo stesso tipo di utente del test 7 modifica correttamente un edificio.	L'utente dev'essere indirizzato alla pagina di amministrazione edifici.	✓
13	Lo stesso tipo di utente del test 7 tenta di modificare un edificio con attributi non validi.	L'utente non dev'essere indirizzato, deve rimanere nella pagina di creazione edificio.	✓
14	Lo stesso tipo di utente del test 7 elimina correttamente un edificio.	L'utente dev'essere indirizzato alla pagina di amministrazione edifici.	✓
da 15 a 20	Un utente autenticato privo del privilegio di modificare edifici richiede l'esecuzione di azioni riservate ad utenti che ne siano in possesso.	L'utente dev'essere indirizzato alla pagina principale.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su ClassroomsController

ID TEST	DESCRIZIONE	OBIETTIVO	ESITO
da 21 a 32 e da 47 a 51	Un utente non autenticato(guest) richiede l'esecuzione di azioni riservate ad utenti con privilegi.	Il guest deve essere indirizzato alla pagina di login.	✓
33	Un utente autenticato con il privilegio di poter modificare le aule richiede l'esecuzione dell'azione new.	L'utente deve essere indirizzato alla pagina di creazione di un'aula.	✓
35	Lo stesso tipo di utente del test 33 crea correttamente una classe.	L'utente deve essere indirizzato alla pagina di amministrazione aule.	✓
36	Lo stesso tipo di utente del test 33 crea una classe con attributi non validi.	L'utente non deve essere indirizzato ma rimanere nella pagina di creazione della classe.	✓
37	Lo stesso tipo di utente del test 33 modifica correttamente una classe.	L'utente deve essere indirizzato alla pagina di amministrazione aule.	✓
38	Lo stesso tipo di utente del test 33 modifica una classe con attributi non validi.	L'utente non deve essere indirizzato ma rimanere nella pagina di creazione della classe.	✓
39	Lo stesso tipo di utente del test 33 elimina una classe.	L'utente deve essere indirizzato alla pagina di amministrazione aule.	✓
44	Lo stesso tipo di utente del test 33 crea un vincolo associato ad una determinata classe.	Deve essere renderizzata la pagina di modifica aule.	✓
45	Lo stesso tipo di utente del test 33 elimina un vincolo associato ad una determinata classe.	Deve essere renderizzata la pagina di modifica aule.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su CurriculumController

ID TEST	DESCRIZIONE	OBIETTIVO	ESITO
da 51 a 58	Un utente non autenticato(guest) richiede l'esecuzione di azioni riservate ad utenti con privilegi.	L'utente dev'essere indirizzato alla pagina di login.	✓
59	Un utente autenticato con privilegio di modificare curricula richiede l'esecuzione dell'azione <code>new</code> .	L'utente dev'essere indirizzato alla pagina di creazione curriculum.	✓
60	Lo stesso tipo di utente del test 59 richiede l'esecuzione dell'azione <code>edit</code> .	L'utente dev'essere indirizzato alla pagina di modifica dei dati relativi ad un curriculum.	✓
61	Lo stesso tipo di utente del test 59 crea correttamente un curriculum.	L'utente dev'essere indirizzato alla pagina di amministrazione curricula.	✓
62	Lo stesso tipo di utente del test 59 tenta di creare un curriculum con attributi non validi.	L'utente non dev'essere indirizzato, deve rimanere nella pagina di creazione curriculum.	✓
63	Lo stesso tipo di utente del test 59 modifica correttamente un curriculum.	L'utente dev'essere indirizzato alla pagina di amministrazione curricula.	✓
64	Lo stesso tipo di utente del test 59 tenta di modificare un curriculum con attributi non validi.	L'utente non dev'essere indirizzato, deve rimanere nella pagina di modifica curriculum.	✓
65	Lo stesso tipo di utente del test 59 richiede l'esecuzione dell'azione <code>edit_teachings</code> .	L'utente dev'essere indirizzato alla pagina di amministrazione degli insegnamenti di un curriculum.	✓
da 66 a 68	Lo stesso tipo di utente del test 59 modifica gli insegnamenti di un curriculum.	L'utente dev'essere indirizzato alla pagina di amministrazione degli insegnamenti di un curriculum.	✓
69	Lo stesso tipo di utente del test 59 elimina correttamente un curriculum.	L'utente dev'essere indirizzato alla pagina di amministrazione curricula.	✓
da 74 a 80	Un utente autenticato privo del privilegio di modificare curricula richiede l'esecuzione di azioni riservate ad utenti che ne siano in possesso.	L'utente dev'essere indirizzato alla pagina principale.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su GraduateCoursesController

ID TEST	DESCRIZIONE	OBIETTIVO	ESITO
da 81 a 86	Un utente non autenticato(guest) richiede l'esecuzione di azioni riservate ad utenti con privilegi.	Il guest deve essere indirizzato alla pagina di login.	✓
91 e 93	Un utente di tipo segreteria(privilegio didactic_office_required) crea correttamente un corso di laurea.	L'utente deve essere indirizzato alla pagina di amministrazione dei corsi di laurea.	✓
92 e 94	Lo stesso tipo di utente del test 91 crea un corso di laurea con attributi non validi.	L'utente deve essere indirizzato alla pagina di amministrazione dei corsi di laurea riportando un errore nella creazione.	✓
95	Un utente autenticato con il privilegio di modificare i corsi di laurea(manage_graduate_courses_required) ne modifica uno.	L'utente deve essere indirizzato alla pagina di amministrazione dei corsi di laurea riportando un messaggio di avvenuta modifica.	✓
96	Lo stesso tipo di utente del test 95 modifica un suo corso di laurea invalidando qualche attributo.	Deve essere renderizzata la pagina di modifica corso di laurea riportando quali errori son stati commessi.	✓
97	Un utente autenticato con i privilegi manage_graduate_courses_required e didactic_office_required elimina un suo corso di laurea.	L'utente deve essere indirizzato alla pagine di amministrazione del corso di laurea.	✓
98	Lo stesso tipo di utente del test 95 tenta di modificare un corso di laurea che non gli appartiene.	L'utente deve essere indirizzato alla pagina principale riportando il seguente messaggio: Non puoi modificare questo corso di laurea.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

ID TEST	DESCRIZIONE	OBIETTIVO	ESITO
100	Lo stesso tipo di utente del test 97 tenta di eliminare un corso di laurea che non gli appartiene.	L'utente deve essere indirizzato alla pagina principale riportando il seguente messaggio: Non puoi modificare questo corso di laurea.	✓
102	Uno user che non è segreteria didattica(quindi non ha il privilegio didactic_office_required) tenta di creare un nuovo corso di laurea.	L'utente deve essere indirizzato alla pagina principale riportando il seguente messaggio: Non possiedi i privilegi per effettuare questa operazione.	✓
103	Uno user che non è segreteria didattica(quindi non ha il privilegio didactic_office_required) tenta di eliminare un corso di laurea.	Lo stesso del test 102.	✓
106	Uno user che non possiede il privilegio di modifica dei corsi di laurea(quindi non ha il privilegio manage_graduate_courses_required) tenta di modificare un corso di laurea.	Lo stesso del test 102.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su SessionsController

ID TEST	DESCRIZIONE	OBIETTIVO	ESITO
108	Un utente non autenticato(guest) richiede l'esecuzione dell'azione new .	L'utente non dev'essere indirizzato, deve rimanere nella pagina di login.	✓
109	Un utente non autenticato(guest) inserisce correttamente username e password.	L'utente dev'essere indirizzato alla pagina principale.	✓
110	Un utente non autenticato(guest) inserisce username e/o password scorretti.	L'utente non dev'essere indirizzato, deve rimanere nella pagina di login.	✓
111	Un utente autenticato richiede l'esecuzione dell'azione destroy .	L'utente dev'essere indirizzato alla pagina di login.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Test su TeachersController

ID TEST	DESCRIZIONE	OBIETTIVO	ESITO
115	Un docente attiva correttamente il proprio user.	L'utente deve essere indirizzato alla pagina principale riportando il seguente messaggio: Account attivato correttamente.	✓
116	Attivazione di un account già esistente	L'utente deve essere indirizzato alla pagina principale riportando il seguente messaggio: L'utente non esiste od è già attivo.	✓
122	Un utente autenticato con il privilegio di modificare docenti (manage_teachers_required) ne invita uno.	L'utente deve essere indirizzato alla pagina principale riportando il seguente messaggio: Docente invitato con successo.	✓
126	Un utente autenticato con il privilegio di modificare privilegi (manage_capabilities_required) ne aggiunge uno ad un docente.	L'utente deve essere indirizzato alla pagina di modifica privilegi.	✓
128	Un utente dello stesso tipo del test 122 assegna ad un docente un corso di laurea.	L'utente deve essere indirizzato alla pagina di modifica corsi di laurea.	✓
129	Un utente dello stesso tipo del test 122 tenta di associare un docente che non appartiene a nessun suo corso di laurea ad un altro corso di laurea.	L'utente deve essere indirizzato alla pagina principale riportando il seguente messaggio: Questo docente non appartiene a nessun tuo corso di laurea.	✓
130	Lo stesso tipo di utente del test 126 tenta di modificare i privilegi di un docente che non appartiene a nessun suo corso di laurea.	Lo stesso del test 129.	✓
131	Lo stesso tipo di utente del test 126 tenta di modificare i privilegi di un docente che non appartiene a nessun suo corso di laurea.	Lo stesso del test 129.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

ID TEST	DESCRIZIONE	OBIETTIVO	ESITO
133	Uno user autenticato come docente(privilegio same_teacher_required) crea una preferenza valida.	Deve essere renderizzata la pagina di modifica preferenza riportando nell'elenco quella appena creata.	✓
134	Lo stesso tipo di user del test 133 crea un vincolo valido.	Deve essere renderizzata la pagina di modifica vincolo riportando nell'elenco quello appena creato.	✓
135	Lo stesso tipo di user del test 133 elimina una preferenza.	Deve essere renderizzata la pagina di modifica preferenza eliminando nell'elenco quella appena cancellata.	✓
136	Lo stesso tipo di user del test 133 elimina un vincolo.	Deve essere renderizzata la pagina di modifica vincolo eliminando nell'elenco quello appena cancellato.	✓
140	Lo stesso tipo di user del test 133 modifica i propri dati personali.	L'utente deve essere indirizzato alla pagina principale riportando il seguente messaggio: Dati personali aggiornati correttamente.	✓
da 142 a 145	Uno user autenticato senza il privilegio di modifica docenti(manage_teachers_required) richiede l'esecuzione di azioni che ne richiedono il possesso.	L'utente deve essere indirizzato alla pagina principale riportando il seguente messaggio: Non possiedi i privilegi per effettuare questa operazione.	✓
da 146 a 147	Uno user autenticato senza il privilegio di modifica privilegi(manage_capabilities_required) richiede l'esecuzione di azioni che ne richiedono il possesso.	L'utente deve essere indirizzato alla pagina principale riportando il seguente messaggio: Non possiedi i privilegi per effettuare questa operazione.	✓

Test su TeachingsController

ID TEST	DESCRIZIONE	OBIETTIVO	ESITO
da 151 a 159	Un utente non autenticato(guest) richiede l'esecuzione di azioni riservate ad utenti con privilegi.	L'utente dev'essere indirizzato alla pagina di login.	✓
161	Un utente autenticato con privilegio di modificare insegnamenti richiede l'esecuzione dell'azione <code>new</code> .	L'utente dev'essere indirizzato alla pagina di creazione insegnamento.	✓
162	Lo stesso tipo di utente del test 161 richiede l'esecuzione dell'azione <code>edit</code> .	L'utente dev'essere indirizzato alla pagina di modifica dei dati relativi ad un insegnamento.	✓
163	Lo stesso tipo di utente del test 161 crea correttamente un insegnamento.	L'utente dev'essere indirizzato alla pagina di selezione docente.	✓
164	Lo stesso tipo di utente del test 161 tenta di creare un insegnamento con attributi non validi.	L'utente non dev'essere indirizzato, deve rimanere nella pagina di creazione insegnamento.	✓
165	Lo stesso tipo di utente del test 161 richiede l'esecuzione dell'azione <code>select_teacher</code> .	L'utente dev'essere indirizzato alla pagina di selezione docente.	✓
166	Lo stesso tipo di utente del test 161 assegna correttamente un docente ad un insegnamento.	L'utente dev'essere indirizzato alla pagina di amministrazione insegnamenti.	✓
167	Lo stesso tipo di utente del test 161 non assegna correttamente un docente ad un insegnamento.	L'utente non dev'essere indirizzato, deve rimanere nella pagina di selezione docente.	✓
168	Lo stesso tipo di utente del test 161 modifica correttamente un insegnamento.	L'utente dev'essere indirizzato alla pagina di amministrazione insegnamenti.	✓



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

169	Lo stesso tipo di utente del test 161 tenta di modificare un insegnamento con attributi non validi.	L'utente non dev'essere indirizzato, deve rimanere nella pagina di modifica insegnamento.	✓
170	Lo stesso tipo di utente del test 161 elimina correttamente un insegnamento.	L'utente dev'essere indirizzato alla pagina di amministrazione insegnamenti.	✓
da 174 a 181	Un utente autenticato privo del privilegio di modificare curricula richiede l'esecuzione di azioni riservate ad utenti che ne siano in possesso.	L'utente dev'essere indirizzato alla pagina principale.	✓

Test su UsersController

ID TEST	DESCRIZIONE	OBIETTIVO	ESITO
da 182 a 183	Un utente non autenticato(guest) richiede l'esecuzione di azioni riservate ad utenti autenticati.	L'utente dev'essere indirizzato alla pagina di login.	✓
184	Un utente autenticato richiede l'esecuzione dell'azione <code>edit</code> .	L'utente viene indirizzato alla pagina di modifica password.	✓
185	Un utente autenticato modifica correttamente la password.	L'utente viene indirizzato alla pagine principale.	✓
186	Un utente autenticato non modifica correttamente la password.	L'utente non viene indirizzato, deve rimanere nella pagina di modifica password.	✓
187	Un utente autenticato richiede l'esecuzione dell'azione <code>destroy</code> .	L'utente viene indirizzato alla pagina di login.	✓

2.3 Metriche

Il progetto 'Sigeol', al termine della fase di programmazione, è stato testato con numerosi strumenti dedicati a Ruby on Rails con lo scopo di generare delle metriche precise che indichino il livello qualitativo del codice prodotto dal team QuiXoft.

E' stato scelto di utilizzare MetricFu 1.0.2, raccolta di gemme per Ruby on Rails scaricabile da <http://metric-fu.rubyforge.org/>

Nei capitoli seguenti saranno analizzate una ad una tutte le metriche utilizzate:

2.3.1 Rcov

Lo strumento Rcov è stato usato per misurare la copertura dei test sul codice. Il team QuiXoft utilizzerà la metrica C0, ovvero sarà controllata la copertura di ogni istruzione, e quindi di ogni riga di codice dell'intero progetto.

La situazione di copertura dei test sul codice è riportato nel file *coverage.pdf* allegato al presente documento.

INSERIRE I RISULTATI DI RCOV

2.3.2 Flay

Qualsiasi progetto Rails dovrebbe seguire il più possibile il principio DRY: Don't Repeat Yourself. Anche se questo deve essere inteso come princi-

pio generale, dovremmo anche evitare di avere ripetizioni all'interno del medesimo file.

Flay permette di tenere sotto controllo le duplicazioni, analizzando le “similarità strutturali” (branch, cicli, etc) presenti nel codice: se due parti di codice sono simili allora potrebbero essere buone candidate per un refactoring.

Il risultato di tale metrica di analisi si può consultare nel file *flay.html* allegato al presente documento.

Analizzando i risultati si possono notare numerose ripetizioni: la maggior parte non sono tuttavia evitabili, in quanto diversi metodi all'interno degli stessi controller hanno un redirect alla stessa pagina, e per forza di cose il codice che effettua il redirect deve essere ripetuto. Nonostante ciò, la quantità di ripetizioni non è stata ritenuta preoccupante dai membri del team QuiXoft, ne sono stati rilevati eventuali problemi per la futura manutenzione del codice.

2.3.3 Flog

Lo strumento Flog applica una metrica ABC al codice del progetto ‘Sigeol’ al fine di misurarne la complessità. La metrica ABC misura la distanza euclidea dall'origine nello spazio tridimensionale formato da:

- Assignments
- Branch
- Condition

Più il codice risulta “lineare” minore sarà il valore di tale metrica ad esso applicato (e tenderà ad essere più gestibile). Al contrario, nel caso di codice ricco di cicli, sottocicli e diversi branch, mostrerebbe un valore elevato, indicando un codice, almeno in teoria, più soggetto a bachi.

Il risultato di tale metrica di analisi si può consultare nel file *flog.html* allegato al presente documento.

La seguente tabella, messa a disposizione degli sviluppatori di Flog, indica la scala di valori con cui misurare la bontà dei risultati, prendendo in esame il punteggio per metodo (seconda colonna del file allegato):

PUNTEGGIO	RISULTATO
0 - 10	Ottimo
11 - 20	Buono
21 - 40	Abbastanza buono (da valutare un possibile refactoring)
41 - 60	Accettabile ma sconsigliato
60 - 100	Pericoloso
100 o più	Non accettabile

Osservando i risultati ottenuti, si può notare che solamente 2 classi rientrano nella fascia 21-40, mentre tutte le altre possono considerarsi buone o ottime. Le 2 classi imputate sono `teacher_controller` e `classroom_controller`: ne è stato valutato un refactoring, ma si è deciso di mantenerle così come sono in quanto la loro complessità è stata ritenuta perfettamente accettabile dai membri del team QuiXoft.

2.3.4 Saikuro

Saikuro è uno strumento per misurare la complessità ciclomatica, metrica strutturale relativa al flusso di controllo di un programma che rappresenta la sua complessità logica, cioè lo sforzo per realizzarlo e comprenderlo.

Il risultato di tale metrica di analisi si può consultare nel file *saikuro.html* allegato al presente documento.

La seguente tabella indica la scala di punteggi con cui valutare i risultati ottenuti:

PUNTEGGIO	RISULTATO
0 - 10	Ottimo
11 - 20	Buono
21 - 40	Sufficiente
41 o più	Insufficiente

Esaminando il file allegato, si può notare come tutti i metodi rientrino nelle prime 2 categorie, indicando una limitata complessità ciclomatica, a tutto vantaggio della facilità di gestione e di manutenzione del codice del progetto SIGEOL.

2.3.5 Reek

Reek è uno strumento di analisi del codice che ci dice se e dove compaiono pattern “sospetti”, come ad esempio:

- metodi troppo lunghi
- classi troppo ampie
- nomi criptici
- liste di parametri eccessivamente lunghe
- duplicazioni

Il risultato di tale metrica di analisi si può consultare nel file *reek.html* allegato al presente documento.

Al termine della fase di programmazione molte parti di codice del progetto ‘Sigeol’ sono state modificate e ottimizzate tenendo conto delle considerazioni nate consultando il risultato di Reek, ma molte delle segnalazioni



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

ancora presenti nel file allegato sono state ritenute poco significative o addirittura in contraddizione con scelte fatte durante lo sviluppo: si è ritenuta quindi irrealizzabile l'idea di eliminare completamente tutti i problemi segnalati da questo strumento.

2.3.6 Roodi

Roodi è l'acronimo di “Ruby Object Oriented Design Inferometer”. E' uno strumento che analizza il nostro codice relativamente a:

- nomi delle classi
- nomi dei metodi
- complessità ciclomatica (sia a livello di metodi che di blocchi)
- blocchi “rescue” vuoti
- cicli del tipo “for”
- lunghezza dei metodi

Il risultato di tale metrica di analisi si può consultare nel file *roodi.html* allegato al presente documento.

Roodi segnala parecchi metodi con lunghezza superiore alle 20 righe: il team QuiXoft ha ritenuto che tale valore sia piuttosto limitante, e per questo ha dato un peso relativamente basso a tale segnalazione. Il numero di righe per metodo non supera mai comunque le 40 righe, valore ritenuto un ottimo compromesso per quanto riguarda la leggibilità e la manutenibilità del codice.

2.3.7 Churn

Churn tiene traccia dei file soggetti a maggiori modifiche. Un file il cui contenuto cambia spesso è un sintomo di probabile necessità di refactoring: forse è necessario rivedere le entità definite, oppure introdurre un diverso design pattern. Churn inferisce i cambiamenti attraverso l'analisi dei log dei Subversion.

Il risultato di tale metrica di analisi si può consultare nel file *churn.html* allegato al presente documento.

Anche in questo caso il `teacher_controller` risulta essere il file modificato più volte: il numero delle modifiche resta comunque accettabile, ed è giustificato dalla notevole complessità delle operazioni che tale controller è tenuto a gestire. Tutti gli altri file sono stati limitatamente modificati, indice di una corretta progettazione.



2 RESOCONTO DELLE ATTIVITÀ DI VERIFICA

Diario delle modifiche

DATA	VERSIONE	MODIFICA
04-06-2009	Delta	Aggiunte le descrizioni dei tool di misurazione delle metriche