

**GUIDA UTENTE**

**Questa guida**

**Lo scopo del presente documento è quello di introdurre l’utente all’utilizzo ed alle logiche su cui si basa il funzionamento di ACDQ Engine. Lo stile scelto è molto discorsivo e basato su esempi allo scopo di essere semplice ma allo stesso tempo comprensibile ed esaustivo. Gli utenti a cui si rivolge sono funzionali tranne l’ultimo paragrafo introduttivo che parla delle API per estendere le funzionalità del prodotto – gli utenti a cui si rivolge in questo caso sono sviluppatori.**

**Introduzione al prodotto**

ACDQ Engine è un prodotto che svolge i seguenti compiti:

* Contenere e gestire informazioni
* Presentarle agli utenti
* Manipolarle

Tecnicamente, è un’applicazione web basata su Java, con un DB MySql – come application Server usa Tomcat.

Le principali caratteristiche sono:

* Altissima flessibilità nella definizione del modello dati
* Modifiche al modello senza necessità di deploy applicativi
* User-friendly
* Basato sul web: non sono necessarie installazioni, manutenzioni e upgrade lato client
* Estendibile: tramite API è possibile aggiungere logica e/o creare applicazioni alternative che utilizzano la medesima base dati
* Presentazione/modifica dei dati: sia in formato web che in formato csv
* Condivisione delle informazioni
* Risoluzione dei permessi/diritti di accesso utente a livello di singola informazione
* Predisposto per interfacce web service Rest e SOAP

**Login**

Per accedere ad ACDQ Engine è necessario inserire 3 parametri

* Un identificativo della società di cui si fa parte
* Un’email che identifica l’untente
* Una password

In caso di riconoscimento di questa tripla di valori sarà possibile accedere al sistema, altrimenti i campi verranno svuotati per eseguire un nuovo tentativo.

Nella home, oltre ai filtri per le ricerche e le altre funzionalità espresse nei paragrafi successivi, è presente una “bacheca” in cui ogni utente può scrivere un messaggio SMS-like (lunghezza 160 caratteri), che diventerà visibile a tutti gli utenti delle società.

**Logout**

Una volta che l’utente preme il tasto logout in una qualunque delle sezioni del sito, la sessione utente viene rimossa e per riutilizzare ACDQ Engine un nuovo Login è necessario.

***NB1****: Login e Logout utente vengono tracciate a livello audit (si veda l’apposita sezione per approfondimenti)*

***NB2****: Gli utenti possono essere amministratori (A) o semplici (U)*

* *Gli amministratori possono creare e modificare campi formula (si veda la sezione relativa), possono creare/modificare/cancellare tutti gli utenti della compagnia di riferimento*

**Campi**

I Campi sono le parti elementari di ACDQ Engine. Ogni utente si può creare la sua collezione di campi che può decidere di usare come vuole. Esempi di campi possono essere nome, cognome, codice fiscale, lista degli alunni di una classe etc. Inoltre, un utente può ereditare campi speciali creati da altri utenti, come verrà spiegato in seguito.

I campi hanno un nome che li identifica, un tipo e, dove previsto una lunghezza massima, un’intestazione ed un valore di default.

Di seguito passiamo in rassegna i tipi nativi con le loro caratteristiche:

* Text: è un comune testo (esempio: cognome, numero di telefono) – ha un nome, lunghezza e un possibile default
* date: è una data (esempio: data di nascita) – ha un nome, lunghezza e un possibile default
* number: è qualunque numero (esempio: numero spettatori) – ha un nome e un possibile default
* choice: è un campo vero/falso (esempio: maggiorenne?) – ha un nome
* select: è un campo scelta multipla (esempio: impiegato, quadro o dirigente?) – ha un nome e un set di possibilità personalizzabili dall’utente
* formula: è un campo calcolato (esempio: IVA) – si ottiene con una query diretta al DB ACDQ Engine che può contenere il parametro :ID: dell’informazione a cui si riferisce – per poter utilizzare questo campo bisogna avere i pemessi di amministratore e conoscere le tabelle interne del database di ACDQ Engine – ha un nome ed una query relativa
* send\_email: è un campo vero/falso che è fisso (ogni informazione creata ne possiederà in automatico uno): serve a dire se mandare notifiche email ogniqualvolta ci sono modifiche alle informazioni oppure no (per approfondimenti si veda la sezione sulla informazioni – ha un nome
* versioning: è un campo vero/falso che è fisso (ogni informazione creata ne possiederà in automatico uno): serve a dire se salvare una versione dell’informazione ad ogni salvataggio per poter avere uno storico di tutte le versioni consultabile in ogni momento
* calculate: è un campo vero/falso che è fisso (ogni informazione creata ne possiederà in automatico uno): serve a dire se eventuali metodi di calcolo sull’informazione devono abilitati o meno
* discussion: questo tipo consente a tutti gli utenti che hanno accesso all’informazione di aggiungere commenti in un formato simile a un forum: questo serve per tener traccia nello stesso punto di commenti degli utenti senza dover ricercare in svariate mail o altre fonti.
  + Ad esempio, se si immagazzina un’attività di un progetto, un campo discussion permette di tenere l’analisi, ripensamenti ecc tutti dentro l’informativa stessa dell’attività
* per i campi text, date, number e formula è presente anche un equivalente lista: list\_text, list\_number, list\_date, list\_formula: questi campi sono pensati per avere più elementi in serie di uno stesso tipo (esempio: lista di attività da svolgere) – hanno un nome
  + il tipo list\_text merita un discorso apparte: possiede infatti anche un’intestazione che, se usata, rende la lista una tabella
    - l’intestazione della tabella ha il seguente formato:
      * nome colonna:tipo colonna:obbligatorio(yes/no)@@ nome colonna:tipo colonna:obbligatorio(yes/no)@@... (“:” e “@@” sono separatori all’occorrenza modificabili)
      * esempio di intestazione - attivita:text:yes@@fine:date:yes
* oltre a tutti questi tipi di campi “nativi” ci sono altri tipi di campi “speciali” che vengono creati dall’utente.
  + La creazione di questi tipi di dato avviene quando l’utente crea una categoria
    - Un esempio: l’utente crea la categoria “dipendente” con i campi nome, cognome e telefono di tipo text
    - La categoria dipendente diventa anche un tipo di dato il che significa che sarà possibile definire dei campi di tipo “dipendente”
    - Si potrebbe entrare nella categoria “dipendente” definita sopra ed aggiungere un campo di tipo “dipendente” di nome “responsabile” – si andrebbe a definire chi è il capo dei dipendenti registrati a sistema: in questo modo il nome del responsabile non è solo un testo con un nome ma è un riferimento ad un altro utente registrato

I campi, se non in uso da almeno una categoria, sono cancellabili

***NB1****: il nome del campo è unico per ogni utente il che significa che:*

* *ogni utente può avere un solo campo cognome*
* *nel sistema ci possono essere più campi cognome appartenenti a più utenti*

***NB2****: il campo è indipendente dalla categoria quindi è usabile in più di esse*

* *ad esempio un campo “nome” si può usare in una categoria “dipendente” (nome del dipendente) ed in una “progetto” (nome del progetto)*

***NB3****: sui campi formula:*

* *inserita la query c’è un processo di validazione della stessa*
  + *in caso il tipo di campo sia* ***“formula”*** *la query deve restituire* ***uno e un solo risultato***
  + *in caso sia di tipo* ***“list\_formula”*** *deve restituire da* ***1 a n*** *risultati*

**Categorie**

Le categorie sono i secondi elementi base di ACDQ Engine: di fatto sono degli insiemi di campi.

Una categoria può essere, come visto sopra “dipendente” con i campi nome, cognome e telefono ad essa associati. Questo significa che ogni volta (come si vedrà in seguito) che verrà creata un’informazione di categoria “dipendente” l’utente potrà assegnare un nome, un cognome ed un telefono.

Ogni categoria ha un nome, un elenco di utenti che diventano “manager” di categoria ed un insieme di campi associati.

Gli ultimi due punti meritano un approfondimento.

**Manager di categoria**

Gli utenti manager di categoria (possono essere più di uno) hanno principalmente quattro possibilità:

* possono creare informazioni di quella categoria
  + le informazioni, salvo condivisione delle stesse come si vedrà in seguito, saranno visibili solo a chi ha creato l’informazione
* possono modificare l’anagrafica della categoria (nome, lista manager, campi associati)
  + queste modifiche saranno visibili a tutti quelli che sono manager di categoria
* possono creare campi del tipo della categoria (come anticipato nella sezione sui campi)
  + l’utente che viene posto a manager della categoria “dipendente” può creare campi di tipo “dipendente”
* possono dare l’ownership della categoria (per poterla modificare a loro volta o creare informazioni) ad altri utenti

**Campi associati**

Nell’associare i campi ad una categoria l’utente può utilizzare tutta la sua collezione di campi a patto che ogni campo appaia solo una volta nella categoria (non posso definire una categoria che contiene due volte un campo di nome “cognome”).

Per i campi **che non sono** “list\_”, “choice”, “formula” ci sono due versioni: obbligatorio (MAN) e non obbligatorio. Per gli obbligatori, al momento di creazione o modifica dell’informazione qualora manchino e vengano svuotati, un errore bloccante eviterà all’utente di proseguire (ad esempio, per un dipendente, il cognome è fondamentale, il terzo recapito telefonico no).

Se l’utente modifica una categoria esistente togliendo un campo questo verrà tagliato da tutte le informazioni di quella categoria causando una potenziale perdita di informazione.

Se invece un campo viene aggiunto, viene messo in automatico a tutte le info precedentemente create di quella categoria (se per il campo è previsto un default viene messo in automatico quel valore).

Un utente che modifica la categoria può togliere un qualunque campo ma può aggiungere solo quelli della sua collezione.

Modificare il nome della categoria vuol dire modificre anche il nome del tipo di campo relativo

* se modifico “dipendente” in “dipendente2” tutti i manager di categoria vedranno il tipo “dipendente2” con effetto retroattivo sui campi precedenti di tipo “dipendente”

**Workflow e nomi automatici**

Creando una categoria, oltre al nome, gli utenti che possono accedervi e la collezione di campi, è possibile definire altre tre caratteristiche:

* se una categoria è workflow:
  + questo significa che gli utenti che possono modificare informazioni di quella categoria possono associare uno “stato” all’informazione, in base al suo stato dell’arte (si pensi ad un’informazione che descrive un progetto). Gli stati possono essere:
  + aperto
  + in attesa (i.e. per un’altra attività bloccante)
  + richiesta di chiarimenti
  + terze parti (il lavoro si è dato ad un’altra società)
  + chiuso (l’attività è stata portata a termine)
    - è sempre possibile passare da uno stato all’altro, tranne tornare ad “aperto” che è lo stato iniziale
    - la storia dei cambiamenti di stato è consultabile
* se un workflow è da approvare prima di partire
  + In questo caso si aggiunge uno stato al workflow: “da approvare”
    - Quando l’utente crea un’informazione di una categoria workflow da approvare significa che, per poter essere chiusa, mandata in terze party, richiesta chiarimento, in attesa va prima approvata dal responsabile dell’utente che l’ha creata
      * Il responsabile è definito in fase di creazione/modifica dell’utente
* Se alle informazioni della categoria si vuole dare un nome automatico
  + Questo significa che, definendo una radice da 1 a 3 caratteri (i.e. “AUT”) è possibile fare in modo che venga proposto un nome in fase di creazione della info (che si può comunque modificare)
    - Il nome è del tipo radice+timestamp (i.e. “AUT121015131224”)

***NB1****: Ogniqualvolta si crea una nuova informazione i campi send\_email, versioning e calculate per attivare/disattivare le modifiche vengono creati in automatico*

***NB2****: il nome della categoria è unico per ogni utente il che significa che:*

* *ogni utente può avere una sola categoria dipendente*
* *nel sistema ci possono essere più categorie dipendente appartenenti a più utenti*

***NB3:*** *In caso di nome automatico, la proposta di questo viene fatta solo nell’import da GUI*

***NB4:*** *gli stati del workflow sono modificabili solo da GUI*

**Informazioni**

Le informazioni sono il cuore di ACDQ Engine, rappresentando di fatto il contenuto informativo degli utenti a differenza di campi e categorie che sono dei componenti di infrastruttura.

Un’informazione si potrebbe definire come un’istanza di una categoria. Ragionando anche qui con esempi, se la categoria è dipendente, informazioni di tipo dipendente sono MarioRossi e AndreaBianchi.

La chiave dell’informazione è il subject – il subject deve essere unico, anche tra più categorie diverse – quindi non può esistere un’informazione MarioRossi di tipo dipendente ed una Mario Rossi di tipo Progetto. Il subject può contenere solo caratteri alphanumerici e “\_”,”-“ – quindi no spazi e altri caratteri.

Nell’inserimento/modifica dell’informazione si tiene conto, come scritto nella sezione sulle categorie, di campi obbligatori e non.

Quando si crea/modifica un’informazione è possibile mettere in scrittura o in sola lettura altri utenti della società, che potranno quindi accedere all’informazione (pur non essendo magari manager di categoria). Gli utenti in sola lettura potranno solo modificare i campi di tipo discussion aggiungendo nuovi commenti in stile forum. **Gli utenti che possono modificare ma che non sono gli autori dell’informazione, non possono cancellarla.**

Alle informazioni possono venire associati dei documenti in allegato (in formato excel, word e power point senza macro, pdf, jpg, bnp, txt e csv) anche dagli utenti in sola lettura.

**Facendo l’upload di immagini jpg che contengono “id\_” nel nome del file, quest’immagine verrà inserita in visualizzazione nel front end dell’informazione. Se ci sono più immagini associate ad un’informazione che contengono “id\_”, viene preso il più recente.**

Le informazioni sono cancellabili

**E’ possibile clonare un’informazione creando un’altra istanza del tutto identica a livello di valore dei campi ed utenti ma col nome=nome+”\_clone”. Ad esempio l’informazione clone di “ristorantedamario” sarebbe “ristorantedamario\_clone”**

**Tabelle e liste**

Come detto sopra, alcuni campi di una categoria possono essere tipi di campo complesso come tabelle e liste. Al momento di inserire elementi di questi campi verranno effettuati gli stessi controlli di integrità dei normali campi. Da interfaccia grafica gli inserimenti di questi elementi sono guidati, nel caso di import/update da csv c’è da seguire una sintassi specifica a cui si rimanda nell’apposita sezione.

**Tasto “considera”**

Gli utenti abilitati alle modifiche su una certa informazione possono premere un tasto di “considerazione” dell’informazione e possono riazzerarlo quando vogliono. Il senso di questo indicatore può essere molteplice e dipende dal tipo di uso che si intende dare. Ad esempio, si può utilizzare per esprimere approvazione o meno nei confronti del contenuto di un’informazione.

**Ricerche**

Visto il numero di informazioni ed il loro tipo, che può crescere anche di tanto, sono state predisposte delle funzionalità avanzate di ricerca.

Ci sono due tipi di filti

**Filtro Generale:** estrae risultati se tutte le condizioni sotto sono verificate

* chiave: tutta o parte del subject delle informazioni da cercare, se lasciata vuota non influisce sulla ricerca
* categoria: permette di filtrare una particolare categoria, se non usato non influisce
* autore: mail di chi l’ha creata, se vuota non influisce
* condivisa: se selezionata permette di estarre solo le informazioni per cui si è in sola lettura, altrimenti non influisce
* stato: lo stato dell’informazione, utile nel caso dei workflow, altrimenti è sempre “aperto”
* azienda: nel caso di accesso ad ACDQ Online, permette di filtrare tra le informazioni per cui si ha il permesso di accedere anche se di altre società, se vuoto non influisce
* versioni: se selezionata permette di trovare tutte i salvataggi precedenti

**Filtro categoria:** estrae anch’esso risultati se tutte le condizioni sotto sono verificate – si attiva una volta selezionata una categoria nel filtro generale. I campi utilizzati sono

* chiave – come per filtro generale
* condivisa – come per filtro generale
* stato – come per filtro generale
* autore – come per filtro generale
* azienda – come per filtro generale
* versioni - come per filtro generale
* NON c’è categoria in quanto specificata nella precedente ricerca
* Tutti campi della categoria specificata (è un filtro DINAMICO, ogni volta viene ricreato da 0, questo garantisce una massima flessibilità nelle ricerche – non sono presenti come filtri campi “list\_” e “formula”)

Nel caso di filtro generale, le ricerche di ogni utente vengono salvate e sono richiamabili in ogni momento. Inoltre, se si specifica una ricerca già fatta ma si forza il valore di un campo, questo viene “forzato” in fase di ricerca. Ad esempio: se l’utente scegli di applicare il filtro già usato in passato “categoria=**automobili**, versioni precedenti=**si**” ma nel campo categoria scrive “moto”, la ricerca verrà fatta applicando il filtro “categoria=**moto**, versioni precedent=**si**”.

**Altre funzionalità**

**Inserimento/aggiornamento da CSV**

Il file csv è un file che contiene un’intestazione (come la “,”) e, nelle righe sotto l’informazione,

Un esempio di estratto di CSV può essere il seguente

***subject,nome,cognome***

***matr23,mario,rossi***

***matr24,andrea,bianchi***

La prima delle 3 righe sopra è l’intestazione, le altre due l’informazione.

ACDQ Engine consente di arricchire o modificare le informazioni usando come ingresso un file csv – l’estensione accettata può essere solo .csv o .txt

Per far si che i csv vengano riconosciuti da ACDQ Engine devono rispettare una certa sintassi

L’intestazione deve avere il seguente formato (le [] indicano parti facoltative)

Subject,[involved],[campo1],[campo2]..

**Ovviamente, il numero delle colonne dovrà essere uguale in ogni riga, altrimenti il file non verrà usato essendo MALFORMATO**

Dalla riga sotto si evince che

* Subject è obbligatorio e deve essere la prima colonna
* Involved non è obbligatorio, ma, se c’è deve essere al secondo posto – questa colonna permette di inserire la lista degli utenti in sola lettura (in condivisione, come visto sopra)
* Campo1.. questi sono i campi che si intende andare a scrivere/aggiornare

Al momento dell’upload del file, vengono controllati i campi inseriti con la categoria selezionata tramite il sito grafico, in particolare si guarda che:

* i caratteri che formano il subject siano solo quelli ammessi (si ricorda che possono essere solo alphanumerici e “\_”,”-“),
* il subject non sia duplicato
* la correttezza di eventuali campi numeri e date e choice (in questo caso i valori nel csv possono essere solo “true”e “false”)
* la correttezza dei campi lista
* **solo in caso di inserimento di una nuova informazione, se mancano campi obbligatori. Una volta superati tutti questi controlli**

Merita attenzione il discorso delle liste.

Nel csv le liste avranno come separatore di riga “;” e come separatore di colonna “@@”

Facendo un esempio, integriamo l’estratto di csv sopra con un nuovo campo, di tipo lista, che indica le sedi di lavoro dei dipendenti:

***subject,nome,cognom,sedi***

***matr23,mario,ross,via rossi@@23@@milano;via bianchi@@34@@roma***

***matr24,andrea,bianch,via rossi@@23@@milano***

**ATTENZIONE ALLE FINE DEL FILE ED ALLA FINE DELLE RIGHE: EVENTUALI SPAZI O RIGHE VUOTE IMPEDIRANNO L’UTILIZZO DEL FILE CHE VERRA’ SEGNALATO COME MALFORMATO**

Da quanto detto sopra si evince che

* l’informazione con subject “matr23” descrive un utente con due sedi di lavoro, “matr24” invece ne ha solo una
* “sedi” è un campo di tipo “list\_text” con intestazione “via:text:yes@@civico:text:yes@@citta:text:yes” (si ricorda che yes/no finale indica se si vuole rendere obbligatorio il campo

***NB1****: qualora una colonna di una lista non fosse obbligatoria e si vuole inserire vuota, nel csv ci dovrà essere il carattere speciale “\_”*

***NB2****: solo in caso di aggiornamento, inserendo prima della lista i caratteri speciali “uuu” (esempio “uuu via rossi@@23@@milano;via bianchi@@34@@roma”) – si va in “append”, ovvero si concatena il contenuto del csv di aggiornamento alla lista già esistente. In caso manchi “uuu” la lista viene CANCELLATA e RICREATA con il solo contenuto del csv*

***NB3****: in caso di aggiornamento, nel csv possono mancare anche campi obbligatori ma, se un campo obbligatorio viene svuotato, l’aggiornamento dell’informazione viene impedito*

**Export**

Con Acdq Engine è possibile esportare in csv le informazioni, si tratta in pratica dell’operazione inversa rispetto allo scorso paragrafo.

Ci sono due modalità di export:

* di una singola categoria (per far questo bisogna entrare nel dettaglio della categoria)
* di tutte le informazioni di una certa categoria (per far questo bisogna introdurre nel filtro di ricerca una categoria specifica)

Il formato dell’export è aderente a quello degli import.

L’export serve essenzialmente a due cose:

* per avere su un file (apribile con un qualsiasi foglio di calcolo ad esempio) le informazioni per fare analisi su di esse
* per importare/modificare da csv (come descritto nel paragrafo sopra) – l’idea qui è modificare ed importare un csv estratto

**Export in excel**

Oltre all’export csv è possibile un export in excel che contiene nel primo sheet tutti i campi semplici, mentre i campi tabella sono posti negli sheet successivi. L’export è pensato per avere in modo semplice una reportistica subito consultabile a differenza del csv che ha scopi più “tecnici”

**Cancellazione di tutte le informazioni di una stessa categoria**

Applicando un filtro specificando la categoria appare la funzionalità che permette di cancellare tutte le informazioni facenti parte di essa. Ad esempio, si possono cancellare tutte le informazioni di tipo “dipendnete”. Questa funzionalità è utile per resettare una categoria ma va usata, ovviamente, con cautela.

**Statistiche**

**E’ presenta una sezione di statistiche che consente di far vedere all’utente le seguenti informazioni**

* **il suo ultimo login**
* quali categorie possiede e può modificare e per le quali può creare informazioni
* quante informazioni hanno creato i dipendenti di cui è responsabile ed in che stato sono (utile per workflow)
* quante informazioni ha creato per ogni categoria ed il loro numero
* in quante informazioni non sue è in scrittura o in sola lettura per ogni categoria
* le sue ultime 1000 operazioni di modifica alla base dati (quindi campi, categorie e informazioni)

**Gestione degli utenti**

Esistono 3 tipi di utenti:

* amministratori (A) : possono fare tutto, compreso creare, modificare e cancellare gli utenti
* utenti (U): possono fare tutto tranne manipolare gli utenti e creare tipi di dato formula
* ospiti (G): possono solo creare e modificare le informazioni per cui sono abilitati – non possono quindi né creare né modificare categorie e campi

*NB1: cancellare gli utenti non significa rimuoverli da ACDQ Engine ma solo disabilitarli, questo ha due effetti*

* *le informazioni e le categorie dell’utente rimosso rimangono disponibili agli altri utenti (anche i suoi campi non vengono rimossi ma essendo solo dell’utente diventano di fatto inutilizzabili)*
* *l’utente cancellato non può più accedere al sistema*

**Metodi API (Per sviluppatori)**

ACDQ Engine rende disponibile una libreria Java (.jar) che contiene un insieme di metodi che consentono di accedere in lettura/scrittura al database andando di fatto ad estendere il prodotto nell’ottica di un framework di sviluppo. In questo modo è possibile creare software che contiene della logica complessa nella creazione/modifica di informazioni (ad esempio software di contabilità o resoconti di analisi chimico/fisiche). In definitiva questa soluzione si rivolge agli ambiti in cui i campi informativi richiedono un’elaborazione particolare per essere popolati.

In questo modo, le possibilità del software sono praticamente infinite consentendo a qualunque società di costruirsi la sua logica in modo del tutto personalizzato. Inoltre, queste funzionalità costituiscono anche una base di partenza per gli export delle informazioni permettendo di creare documenti più complessi e personalizzati rispetto agli export csv nativi di cui si è discusso nella relativa sezione.

Per avere i dettagli dei metodi API disponibili nella release 1.0 si veda il relativo javadoc