

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.

Documentazione tecnica

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

1. INTRODUZIONE	3
2. LIFERAY PORTAL COMMUNITY	3
2.1. INTRODUZIONE	3
2.2. FUNZIONALITÀ PRINCIPALI.....	3
2.3. INFRASTRUTTURA NECESSARIA.....	3
2.4. CONFIGURAZIONE	3
2.5. CARATTERISTICHE TECNICHE.....	5
3. CAS	5
3.1. INTRODUZIONE	5
3.2. FUNZIONALITÀ PRINCIPALI.....	5
4. PORTALE UPI E ISTANZE PROVINCE	6
4.1. INTEGRAZIONE CON CAS.....	6
4.1.1. <i>Completezza dell'anagrafica utente</i>	6
4.2. MODULI ORIZZONTALI	6
4.2.1. <i>PiCo - Entità del database</i>	7
4.3. MODULI VERTICALI.....	22
4.3.1. <i>SIT</i>	22
4.3.2. <i>SAPE</i>	22
4.3.3. <i>SUA</i>	22
4.3.4. <i>SUA - Entità del database</i>	22
5. MODULO FORMAZIONE	53
5.1. INTRODUZIONE	53
5.2. LA SOLUZIONE	53
5.3. INFRASTRUTTURA NECESSARIA	53
5.3.1. <i>Server</i>	53
5.3.2. <i>Sistema operativo</i>	54
5.3.3. <i>Database</i>	54
5.3.4. <i>Web server</i>	54
5.3.5. <i>PHP</i>	54
5.3.6. <i>Software aggiuntivo</i>	54
5.4. FUNZIONALITÀ PRINCIPALI.....	55
5.4.1. <i>Multi-tenancy</i>	55
5.4.2. <i>Gestione centralizzata</i>	55
5.4.3. <i>Condivisione delle risorse</i>	55
5.4.4. <i>Sicurezza</i>	55
5.5. CARATTERISTICHE TECNICHE.....	55
5.5.1. <i>Architettura modulare</i>	55
5.5.2. <i>API REST</i>	56
5.5.3. <i>Integrazione con Liferay Portal</i>	56
5.5.4. <i>Integrazioni</i>	56
5.6. GESTIONE DELLE ISTANZE SU IOMAD	56
5.6.1. <i>Creazione di nuove istanze</i>	56
5.6.2. <i>Clonazione di istanze</i>	57
5.6.3. <i>Aggiornamenti</i>	57
5.6.4. <i>Backup e ripristino</i>	57
5.7. AMMINISTRAZIONE DI IOMAD	57
5.7.1. <i>Creazione corsi</i>	57
5.7.2. <i>Abilitazione corso agli utenti</i>	58
6. HELPDESK (OTRS)	58
6.1. CREAZIONE DEI TICKET.....	59
6.1.1. <i>Modalità di Creazione</i>	59
6.1.2. <i>Campi e Informazioni Gestite</i>	60
6.1.3. <i>Gestione e Aggiornamento dei Ticket</i>	60
6.1.4. <i>Risoluzione e Chiusura del Ticket</i>	61
6.2. INTRODUZIONE ALLA CLASSIFICAZIONE DEI TICKET	61
6.2.1. <i>Elementi Chiave della Classificazione</i>	61
6.2.2. <i>Modalità Operative per la Classificazione dei Ticket</i>	62

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.

nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

6.3. INTRODUZIONE ALL'AUTOMAZIONE DEI WORKFLOW IN OTRS/ZNUNY	63
6.3.1. Componenti di un Workflow Automatizzato	63
6.3.2. Modalità Operative per l'Automazione dei Workflow	64
6.3.3. Esempi Pratici di Automazione dei Workflow	65
6.4. INTRODUZIONE ALLE FAQ IN OTRS/ZNUNY	66
6.4.1. Cosa si può fare con le FAQ in OTRS/ZNUNY	66
6.4.2. Modalità Operative per Creare e Gestire le FAQ	66
6.4.3. Best Practices per una Gestione Efficace delle FAQ	67
6.5. INTRODUZIONE ALLA REPORTISTICA IN OTRS/ZNUNY	68
6.5.1. Cosa Si Può Fare con la Reportistica in OTRS/ZNUNY	68
6.5.2. Modalità Operative per Generare e Gestire i Report in OTRS/ZNUNY	69
6.5.3. Best Practices per la Reportistica in OTRS/ZNUNY	69
6.6. REQUISITI HARDWARE	69
6.6.1. Processore (CPU)	69
6.6.2. Memoria (RAM)	70
6.6.3. Spazio su Disco	70
6.6.4. Connettività e Rete	70
6.7. REQUISITI SOFTWARE	70
6.7.1. Sistema Operativo	70
6.7.2. Server Web	70
6.7.3. Linguaggio di Programmazione	71
6.7.4. Database	71
6.7.5. Altre Componenti e Librerie	71
6.8. INSTALLAZIONE	71
6.9. CONFIGURAZIONE DEL DATABASE	73
6.10. CONFIGURAZIONE DEL SERVER WEB	74
6.11. CONFIGURAZIONE DI AVVIO	74
6.12. CREAZIONE DEGLI UTENTI	75
6.13. CONFIGURAZIONE DEI WORKFLOW	76
6.13.1. Code	76
6.13.2. Servizi	76
6.13.3. SLA (Service Level Agreements)	77
6.13.4. Notifiche	77
6.14. RIFLESSIONI FINALI E PROSPETTIVE FUTURE	78
7. BUSINESS INTELLIGENCE (APACHE SUPERSET)	79
7.1. INTRODUZIONE	79
7.2. FUNZIONALITÀ PRINCIPALI	79
7.3. INFRASTRUTTURA NECESSARIA	80
7.4. INSTALLAZIONE E CONFIGURAZIONE	80
7.5. VANTAGGI DI UTILIZZARE SUPERSET	81
7.6. APPROFONDIMENTI RELATIVI ALLA SOLUZIONE PER UPI	81
7.6.1. Disegno architettonico	81
7.6.2. Specifiche e lista server	82
7.6.3. Setup Server	82
7.6.4. Azure DevOps repository	84
7.6.5. Deploy soluzione Apache Superset	84
7.6.6. Approccio Virtual Machine Scale Set (in AWS)	88
7.6.7. Dettagli Architettonici:	89
7.6.8. Connessione ai dati UPI	90
8. INFRASTRUTTURA AWS	90
8.1. MOODLE TARGET GROUP	90
8.2. LIFERAY TARGET GROUP	91
8.3. OTRS/HELPDESK TARGET GROUP	91
8.4. SUPERSET TARGET GROUP	91
8.5. SIT TARGET GROUP	91
8.6. ALTRI COMPONENTI	91

1. INTRODUZIONE

Quella che segue è la documentazione tecnica sulle varie componenti software dell'applicativo Pi.Co., realizzato nell'ambito del Progetto di UPI “Le Province e il sistema dei servizi a supporto dei Comuni”. Per i necessari approfondimenti sulla parte funzionale, si rimanda al documento “Progetto esecutivo”, prodotto a seguito dei diversi incontri di approfondimento con i principali *stakeholder* istituzionali e validato da UPI per l'avvio degli sviluppi della applicazione.

2. LIFERAY PORTAL COMMUNITY

2.1. INTRODUZIONE

Liferay Portal Community Edition 7 è una piattaforma enterprise open-source che consente di creare portali digitali personalizzati e altamente personalizzabili. Offre un'ampia gamma di funzionalità per la gestione dei contenuti, la collaborazione e la creazione di comunità online.

2.2. FUNZIONALITÀ PRINCIPALI

- **Gestione dei contenuti:** Consente la creazione e la gestione di contenuti web (articoli, blog, pagine) in modo intuitivo.
- **Personalizzazione:** Offre strumenti per personalizzare l'aspetto e la funzionalità del portale, adattandolo alle esigenze specifiche dell'organizzazione.
- **Collaborazione:** Supporta la collaborazione tra utenti attraverso forum, wiki, blog e strumenti di messaggistica.
- **Integrazione:** Si integra con una vasta gamma di sistemi esterni, come CRM, ERP e sistemi di autenticazione.
- **Sicurezza:** Offre funzionalità di sicurezza avanzate per proteggere i dati e le applicazioni.
- **Mobile-first:** Progettato per essere responsive e offrire una buona esperienza utente su dispositivi mobili.

2.3. INFRASTRUTTURA NECESSARIA

- **Server:** Un server con risorse sufficienti per gestire il carico di lavoro previsto.
- **Sistema operativo:** Linux (Ubuntu, CentOS, Debian) è il sistema operativo più comunemente utilizzato.
- **Database:** Un database relazionale come MySQL, PostgreSQL o Oracle per memorizzare i dati del portale.
- **Java:** Liferay richiede l'installazione di una versione specifica di Java.
- **Web server:** Un web server come Tomcat o Jetty per servire le applicazioni web.
- **Applicazione server:** Liferay è un'applicazione Java, quindi è necessario un contenitore di applicazioni come.

2.4. CONFIGURAZIONE

- **Installazione:** Liferay può essere installato manualmente o utilizzando strumenti di automazione come Docker.

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

- **Configurazione del database:** Configurare la connessione al database per memorizzare i dati del portale.
- **Configurazione dell'applicazione server:** Configurare il contenitore di applicazioni per eseguire Liferay.

Configurazione iniziale: Configurare le impostazioni di base del portale, come la lingua, la valuta e il fuso orario.

- **Creazione di utenti e ruoli:** Creare gli utenti e assegnargli i ruoli appropriati in base ai loro permessi.
- **Personalizzazione dell'interfaccia:** Personalizzare l'aspetto del portale utilizzando i temi e i layout predefiniti o creando temi personalizzati.

2.5. CARATTERISTICHE TECNICHE

- **Architettura modulare:** Liferay è costruito su un'architettura modulare che consente di estenderne le funzionalità tramite moduli aggiuntivi.
- **API REST:** Fornisce un'API RESTful per l'integrazione con altri sistemi.
- **Supporto multi-lingua:** Possibilità di creare portali multilingue.
- **Integrazione con LDAP:** Supporta l'autenticazione tramite LDAP per una gestione centralizzata degli utenti.

3. CAS

3.1. INTRODUZIONE

SSO 3PItalia è una soluzione SaaS multitenant basata su Apereo CAS, progettata per consentire a 3PItalia di aggregare gli enti e fornire autenticazione SPID e CIE. Apereo CAS è un Single Sign-On

(SSO) open-source ampiamente adottato per la gestione dell'autenticazione centralizzata, supportando protocolli come CAS, SAML2 e OAuth2.

Il sistema permette agli enti aderenti di integrare i propri servizi con un'infrastruttura scalabile e sicura, garantendo conformità agli standard nazionali di identità digitale. Grazie alla sua architettura modulare, SSO 3PItalia offre elevata interoperabilità e supporto a diversi provider di autenticazione.

Questa soluzione semplifica la gestione delle identità digitali per la Pubblica Amministrazione e gli enti privati, riducendo la complessità di integrazione e migliorando l'esperienza utente.

3.2. FUNZIONALITÀ PRINCIPALI

Ecco le funzionalità principali di **SSO 3PItalia**:

- Autenticazione centralizzata basata su Apereo CAS, con supporto per SPID e CIE.
- Compatibilità con più protocolli: CAS Client, SAML2 e OAuth2, per una facile integrazione con applicazioni esistenti.
- Architettura SaaS multitenant, che consente la gestione di più enti con un'unica piattaforma e la personalizzazione dell'interfaccia grafica in linea con la brand identity dell'ente.
- Gestione sicura delle identità digitali, con crittografia avanzata e conformità agli standard di sicurezza.
- Alta configurabilità per adattarsi alle esigenze specifiche di ciascun ente aggregato.
- Monitoraggio e logging avanzato, con strumenti per il tracciamento degli accessi e la diagnostica degli eventi.
- Interfaccia amministrativa intuitiva, per la gestione centralizzata di utenti, permessi e configurazioni.
- Scalabilità e affidabilità, grazie all'uso di tecnologie cloud-native per garantire prestazioni elevate.

Integrazione con sistemi legacy e moderni, facilitando la transizione verso un modello di autenticazione federata.

Gli applicativi della soluzione Pi.Co. sono integrati con il tenant di riferimento *upi-ss0.3pitalia.cloud* configurato inizialmente e specifico per il progetto, non sono quindi necessari interventi successivi per garantire l'autenticazione ai vari moduli verticali e orizzontali della piattaforma.

È stata infine predisposta un'area LDAP dedicata al progetto utilizzata dai vari applicativi per le seguenti finalità:

- Dati di registrazione degli utenti
- Gruppi di appartenenza ai moduli verticali

4. PORTALE UPI E ISTANZE PROVINCE

Il portale di UPI è costituito da una sola istanza principale.

Al suo interno, sul sito di default (Guest) è stato creato il portale nazionale di UPI mentre, per ogni sito di Provincia, è presente un relativo Sito all'interno della stessa istanza.

4.1. INTEGRAZIONE CON CAS

La piattaforma Liferay offre nativamente la funzionalità di integrazione con il modulo CAS.

Al momento del login, l'utente viene rediretto al CAS e, dopo essersi autenticato, l'utente viene federato direttamente sulla piattaforma Liferay.

4.1.1. Completezza dell'anagrafica utente

Nel caso in cui l'utente si sia registrato partendo dal portale Liferay, allora dopo aver effettuato il login sul modulo CAS sarà automaticamente loggato anche sul portale Liferay.

Nel caso in cui invece l'utente avesse fatto il primo login, senza essersi preventivamente registrato sul portale Liferay, utilizzando i meccanismi di SPID o CIE, allora il portale gli proporrà di modificare la propria anagrafica completando i campi che mancano al suo funzionamento.

4.2. MODULI ORIZZONTALI

Il progetto prevede diversi moduli definiti "orizzontali" perché presenti su tutti i siti di provincia. Di seguito ne illustriamo le caratteristiche e i punti di contatto con la piattaforma Liferay:

- *Documentale*

Per il modulo documentale si è utilizzata la componente "Document & Media" di prodotto. Lato front end è stato utilizzato il widget fornito "out-of-the-box" dalla piattaforma direttamente sulla pagina.

- *Sondaggi*

Per il modulo sondaggi si è utilizzata la componente "Sondaggi" di prodotto. Lato front end è stato utilizzato un widget sviluppato custom che legge le tabelle dell'oggetto standard e ne presenta una lista filtrata sull'utente.

- *Blog*

Per il modulo documentale si è utilizzata la componente "Blog" di prodotto.

Lato front end è stato utilizzato il widget fornito out-of-the-box dalla piattaforma direttamente sulla pagina.

Forum

Per il modulo documentale si è utilizzata la componente “Forum” di prodotto.

Lato front end è stato utilizzato il widget fornito out-of-the-box dalla piattaforma direttamente sulla pagina.

- **Newsletter**

Per il modulo Newsletter è stato creato un applicativo specifico; si veda il file relativo alla definizione delle tabelle del progetto Pi. Co.

4.2.1. PiCo - Entità del database

In questa sezione sono descritte le entità del modulo generale PiCo, ossia quelle di valenza generale per tutti gli altri moduli.

- **Struttura dell'entità ActivityLog**

Rappresenta l'elenco dei log delle singole attività che vengono svolte sul sistema.

Nome	Tipo	Obbligato	Note
activityId	long	Si	Chiave primaria
companyId	long	Si	Audit field
groupId	long	Si	Audit field
userId	long	Si	Audit field
userName	String	Si	Audit field
createDate	Date	Si	Audit field
modifiedDate	Date	Si	Audit field
message	String	Si	Log descrittivo della comunicazione
notes	String	No	Testo aggiuntivo della comunicazione Viene utilizzato per salvare i commenti ai documenti di gara
type	int	Si	Tipologia della comunicazione Si veda la classe <code>it.dvel.comdata.upi.pico.constants.ActivityLogType</code>
classNameId	long	No	Il <code>classNameId</code> della tabella di riferimento per il messaggio.
classPK	long	No	La chiave primaria dell'oggetto che riferisce questa tabella.

Struttura dell'entità Authority

Rappresenta l'elenco degli enti/organizzazioni

Nome	Tipo	Obbligatorio	Note
authorityId	long	Sì	Chiave primaria
companyId	long	Sì	Audit field
userId	long	Sì	Audit field
userName	String	Sì	Audit field
createDate	Date	Sì	Audit field
modifiedDate	Date	Sì	Audit field
code	String	Sì	Codice univoco dell'ente
name	String	Sì	Nome dell'ente
roleId	long	no	Identificativo del ruolo regolare associato all'ente
province	boolean	no	Stabilisce se l'ente rappresenta una provincia
provinceCode	String	Sì	Codice univoco della provincia Dato ridondato da City
referenceProvinceCode	String	Sì	Codice univoco della provincia di riferimento Rappresenta la provincia, il cui responsabile riceverà la notifica di approvazione della registrazione utente; il suo valore iniziale è uguale a provinceCode.
fiscalCode	String	Sì	Codice fiscale dell'ente
authorityTypeId	long	Sì	Tipo dell'ente
cityId	long	Sì	Città dell'ente
zipCode	String	Sì	CAP specifico dell'ente
address	String	Sì	Indirizzo dell'ente
pecEmailAddress	String	Sì	Indirizzo PEC dell'ente

Indici univoci

Campi		Descrizione
companyId, code	Non possono esistere 2 sso enti con lo ste	codice

Struttura dell'entità AuthorityType

Rappresenta l'elenco delle tipologie di ente.

Nome	Tipo	Obbligatorio	Note
authorityTypeId	long	Sì	Chiave primaria
companyId	long	Sì	Audit field
userId	long	Sì	Audit field
userName	String	Sì	Audit field
createDate	Date	Sì	Audit field
modifiedDate	Date	Sì	Audit field
code	String	Sì	Codice univoco del tipo di ente
name	String	Sì	Nome del tipo di ente
type	String	Sì	Tipologia/raggruppamento del tipo di ente
sfe	String	Sì	
utd	String	Sì	
nso	String	Sì	
aoo	String	Sì	
uo	String	Sì	

Indici univoci

Campi	Descrizione
companyId, code	Non possono esistere 2 tipi con lo stesso codice

- **Struttura dell'entità AuthorizedProvince**

Rappresenta l'elenco delle province abilitate all'uso del portale; al momento il censimento è manuale da db.

Nome	Tipo	Obbligatorio	Note
authorizedProvinceId	long	Sì	Chiave primaria
companyId	long	Sì	Audit field
userId	long	Sì	Audit field
userName	String	Sì	Audit field
createDate	Date	Sì	Audit field

modifiedDate	Date	Sì	Audit field
liferayGroupId	long	Sì	Identificativo del sito Liferay che rappresenta la provincia
authorityId	long	Sì	Identificativo dell'ente che rappresenta la provincia
code	String	Sì	Codice univoco della provincia Vedi <code>City.provinceCode</code>
name	String	Sì	Nome della provincia Vedi <code>City.provinceName</code> Dato ridondato
data	boolean	No	Abilitazione del modulo dati
helpdesk	boolean	No	Abilitazione del modulo assistenza
moodle	boolean	No	Abilitazione del modulo Moodle
sape	boolean	No	Abilitazione del modulo SAPE
sit	boolean	No	Abilitazione del modulo SIT
sua	boolean	No	Abilitazione del modulo SUA

Indici univoci

Campi	Descrizione
companyId, code	La stessa provincia non può essere autorizzata più volte

- **Struttura dell'entità City**

Rappresenta l'elenco delle città italiane; elenco denormalizzato perché contiene anche i dati di province e regioni.

Nome	Tipo	Obbligatorio	Note
cityId	long	Sì	Chiave primaria
companyId	long	Sì	Audit field
userId	long	Sì	Audit field
userName	String	Sì	Audit field
createDate	Date	Sì	Audit field

modifiedDate	Date	Sì	Audit field
name	String	Sì	Nome della città
provinceCode	String	Sì	Codice della provincia
provinceName	String	Sì	Nome della provincia
regionCode	String	Sì	Codice della regione
regionName	String	Sì	Nome della regione
type	int	Sì	Tipo della città I valori ammessi sono censiti nella classe CityType
capital	boolean	Sì	Determina se si tratta di capoluogo
plateCode	String	Sì	Sigla della targa automobilistica
istatCode	int	Sì	Codice ISTAT
cadastralCode	String	Sì	Codice catastale
nuts1	String	Sì	Codice NUTS1
nuts2	String	Sì	Codice NUTS2
nuts3	String	Sì	Codice NUTS3
provinceCodeISO3166	String	Sì	Codice ISO3166 della provincia
regionCodeISO3166	String	Sì	Codice ISO3166 della regione

Indici univoci

Campi	Descrizione
companyId, istatCode	Non possono esistere 2 città con lo stesso codice ISTAT
companyId, cadastralCode	Non possono esistere 2 città con lo stesso codice catastale

- **Struttura dell'entità Comment**

Rappresenta il singolo commento associato a un'entità.

Nome	Tipo	Obbligatorio	Note
commentId	long	Sì	Chiave primaria
companyId	long	Sì	Audit field

groupId	long	Si	Audit field
userId	long	Si	Audit field Autore del commento
userName	String	Si	Audit field
createDate	Date	Si	Audit field
modifiedDate	Date	Si	Audit field
message	String	Si	Testo del commento
classNameId	long	No	Il <code>classNameId</code> della tabella di riferimento per il messaggio
classPK	long	No	La chiave primaria dell'oggetto che riferisce questa tabella

- **Struttura dell'entità Communication**

Rappresenta l'elenco delle comunicazioni inviate dai singoli moduli.

Nome	Tipo	Obbligatorio	Note
communicationId	long	Si	Chiave primaria
companyId	long	Si	Audit field
groupId	long	Si	Audit field
userId	long	Si	Audit field
userName	String	Si	Audit field
createDate	Date	Si	Audit field
modifiedDate	Date	Si	Audit field
liferayUserId	long	Si	Identificativo dell'utente che ha creato la comunicazione (mittente) Tipicamente sarà lo stesso valore di <code>userId</code>
liferayUserName	String	Si	Rappresenta il nome dell'utente che ha creato la comunicazione (mittente). Dato ridonato dalla tabella User per efficienza query di visualizzazione.
subject	String	Si	Oggetto della comunicazione
description	String	Si	Descrizione della comunicazione

activityLogId	long	Si	Identificativo dell'attività di riferimento
html	boolean	No	Determina se la comunicazione è HTML o testo semplice
fileEntryIds	String	No	Identificativi dei file da inviare come allegati della mail
fileVersions	String	No	Versioni dei file da inviare come allegati della mail
sentDate	Date	No	Data di invio della comunicazione Vale null se non è ancora stata inviata

Indici univoci

Campi	Descrizione
groupId, activityLogId	Non possono esistere 2 comunicazioni per la stessa attività

- **Struttura dell'entità CommunicationRecipient**

Rappresenta la relazione molti a molti tra le comunicazioni e gli utenti del sistema.

Nome	Tipo	Obbligatorio	Note
communicationRecipientId	long	Si	Chiave primaria
companyId	long	Si	Audit field
groupId	long	Si	Audit field
userId	long	Si	Audit field
userName	String	Si	Audit field
createDate	Date	Si	Audit field
modifiedDate	Date	Si	Audit field
communicationId	long	Si	FK della tabella Communication
liferayUserId	long	Si	Id dell'utente destinatario della comunicazione
liferayUserName	String	Si	Rappresenta il nome dell'utente destinatario della comunicazione Dato ridondato dalla tabella User per efficienza query di visualizzazione.

- **Struttura dell'entità Manager**

Rappresenta il singolo responsabile di provincia o comune.

Nome	Tip	Obbligatorio	Note
managerId	long	Sì	Chiave primaria
companyId	long	Sì	Audit field
userId	long	Sì	Audit field
userName	String	Sì	Audit field
createDate	Date	Sì	Audit field
modifiedDate	Date	Sì	Audit field
liferayUserId	long	Sì	Identificativo dell'utente Liferay I dati aggiuntivi sono reperibili dalla tabella <code>UserInfo</code>

type	int	Sì	Determina se il responsabile è della provincia o del comune (o dei tipi che arriveranno in futuro) I valori ammessi sono censiti nella classe <code>ManagerType</code>
code	String	Sì	Codice della provincia o del comune

- **Struttura dell'entità Newsletter**

Rappresenta la singola newsletter inviata dal portale.

Nome	Tipo	Obbligatorio	Note
newsletterId	long	Sì	Chiave primaria
companyId	long	Sì	Audit field
userId	long	Sì	Audit field
userName	String	Sì	Audit field
createDate	Date	Sì	Audit field
modifiedDate	Date	Sì	Audit field
subject	String	Sì	Oggetto della mail
body	String	Sì	Corpo della mail Potrebbe anche essere un semplice template con placeholder
html	boolean	No	Mail in formato HTML oppure no

senderName	String	Sì	Nome del mittente
senderEmailAddress	String	Sì	Indirizzo del mittente
sentDate	Date	No	Data di invio della newsletter Dal momento che sarà inviata una mail per ogni destinatario, questo campo rappresenta la data in cui inizia l'invio ossia quando transita nello stato <code>PARTIAL</code>
status	int	Sì	Stato della newsletter <ul style="list-style-type: none"> • <code>DRAFT</code>, stato iniziale prima dell'invio • <code>PARTIAL</code>, stato intermedio durante l'invio • <code>SENT</code>, stato finale dopo l'invio con successo a tutti i destinatari • <code>ERROR</code>, se si verifica un errore

Gestione degli stati

Lo stato `DRAFT` viene impostato al momento della creazione della newsletter e resta tale fino all'invio; così l'utente può modificare le cose finché è necessario.

Nel momento in cui si decide di inviare la newsletter, l'elemento transita nello stato `PARTIAL` e viene impostato il campo `sentDate`.

Quanto per tutti i destinatari è stata inviata con successo la mail, si transita nello stato `SENT`; in caso di errore di uno o più invii, si transita nello stato `ERROR`.

- **Struttura dell'entità NewsletterRecipient**

Rappresenta il singolo destinatario della newsletter inviata dal portale.

Nome	Tipo	Obbligatorio	Note
newsletterRecipientId	long	Sì	Chiave primaria
companyId	long	Sì	Audit field
userId	long	Sì	Audit field
userName	String	Sì	Audit field
createDate	Date	Sì	Audit field
modifiedDate	Date	Sì	Audit field
newsletterId	long	Sì	Identificativo della newsletter

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

body	String	Sì	Corpo della mail Si tratta del testo effettivamente inviato all'utente, sostituendo eventuali placeholder Questo campo resta vuoto finché la newsletter non viene inviata
liferayUserId	long	Sì	Identificativo dell'utente di portale a cui inviare la mail
fullName	String	Sì	Nome del destinatario Dato ridondato per semplificare l'invio della mail e mantenere lo storico
emailAddress	String	Sì	Indirizzo del destinatario Dato ridondato per semplificare l'invio della mail e mantenere lo storico
sentDate	Date	No	Data di invio della mail
status	int	Sì	Stato del singolo destinatario <ul style="list-style-type: none"> • DRAFT, stato iniziale prima dell'invio • SENT, stato finale dopo l'invio con successo • ERROR, se si verifica un errore
statusMessage	String	No	Messaggio di errore

Struttura dell'entità SupersetUser

Rappresenta l'elenco delle utenze tecniche di accesso alla dashboard dei dati (Superset).

Nome	Tipo	Obbligatorio	Note
supersetUserId	long	Sì	Chiave primaria
companyId	long	Sì	Audit field
userId	long	Sì	Audit field
userName	String	Sì	Audit field
createDate	Date	Sì	Audit field
modifiedDate	Date	Sì	Audit field
screenName	String	Sì	Nome utente
emailAddress	String	Sì	Indirizzo di posta elettronica
firstName	String	Sì	Nome
lastName	String	Sì	Cognome
password	String	Sì	Password

Indici univoci

Campi	Descrizione
companyId, screenName	Non possono esistere 2 utenti con lo stesso nome

- **Struttura dell'entità Survey**

Rappresenta il singolo sondaggio della community.

Nome	Tipo	Obbligatorio	Note
surveyId	long	Sì	Chiave primaria
companyId	long	Sì	Audit field
groupId	long	Sì	Audit field
userId	long	Sì	Audit field
userName	String	Sì	Audit field
createDate	Date	Sì	Audit field

modifiedDate	Date	Sì	Audit field
name	String	Sì	Nome del sondaggio
authorityId	long	Sì	Ente dell'utente creatore
formInstanceId	long	Sì	Identificativo del form
expirationDate	Date	Sì	Data di scadenza
status	int	Sì	Stato del sondaggio

Indici univoci

Campi	Descrizione
groupId, formInstanceId	Non possono esistere 2 sondaggi per lo stesso form

• **Struttura dell'entità SurveyRecipient**

Rappresenta il singolo destinatario del sondaggio.

Nome	Tipo	Obbligatorio	Note
surveyRecipientId	long	Sì	Chiave primaria
companyId	long	Sì	Audit field
groupId	long	Sì	Audit field
userId	long	Sì	Audit field
userName	String	Sì	Audit field
createDate	Date	Sì	Audit field
modifiedDate	Date	Sì	Audit field
surveyId	long	Sì	Identificativo del sondaggio
liferayUserId	long	Sì	Identificativo dell'utente di portale a cui inviare il sondaggio
fullName	String	Sì	Nome del destinatario Dato ridondato per semplificare l'invio della mail e mantenere lo storico
emailAddress	String	Sì	Indirizzo del destinatario Dato ridondato per semplificare l'invio della mail e mantenere lo storico

voteDate	Date	No	Data di votazione del sondaggio
----------	------	----	---------------------------------

Indici univoci

Campi	Descrizione
groupId, surveyId, liferayUserId	Un utente può comparire solamente una volta in un sondaggio

Struttura dell'entità UserInfo

Rappresenta le informazioni aggiuntive legate al singolo utente.

Nome	Tipo	Obbligatorio	Note
userId	long	Sì	Chiave primaria
companyId	long	Sì	Audit field
createDate	Date	Sì	Audit field
modifiedDate	Date	Sì	Audit field
fiscalCode	String	Sì	Codice fiscale
role	String	Sì	Ruolo
area	String	No	Area
office	String	No	Ufficio
authorityId	long	Sì	Identificativo dell'organizzazione/ente di appartenenza
provinceCode	String	Sì	Codice univoco della provincia di appartenenza Dato ridondato ricavabile da Authority.referenceProvinceCode
phoneNumber	String	Sì	Numero di telefono
rup	boolean	No	Determina se l'utente è RUP oppure no
newsletter	boolean	No	Accettazione della newsletter
privacy	boolean	Sì	Accettazione dell'informativa privacy
privacyDate	Date	Sì	Data dell'accettazione

- **Struttura dell'entità UserRequest**

Rappresenta la singola richiesta di approvazione della registrazione di un utente.

Nome	Tipo	Obbligatorio	Note
userRequestId	long	Si	Chiave primaria
companyId	long	Si	Audit field
createDate	Date	Si	Audit field
modifiedDate	Date	Si	Audit field
provinceCode	String	Si	Codice della provincia che dovrà approvare la richiesta Dato ridondato ricavabile da Authority.referenceProvinceCode
firstName	String	Si	Nome dell'utente
lastName	String	Si	Cognome dell'utente
emailAddress	String	Si	Email dell'utente
birthday	Date	Si	Data di nascita dell'utente
fiscalCode	String	Si	Codice fiscale dell'utente
role	String	Si	Ruolo dell'utente
area	String	No	Area
office	String	No	Ufficio
authorityId	long	Si	Identificativo dell'organizzazione/ente di appartenenza
phoneNumber	String	Si	Telefono dell'utente
services	String	Si	Servizi da abilitare (SUA, SAPE, ...)
newsletter	boolean	No	Accettazione della newsletter
privacy	boolean	Si	Accettazione dell'informativa privacy
status	int	Si	Stato della richiesta I valori ammessi sono censiti nella classe UserRequestStatusConstants
statusByUserId	long	Si	Identificativo dell'utente che ha cambiato lo stato
statusByUsername	String	Si	Nome completo dell'utente che ha cambiato lo stato
statusDate	Date	Si	Data in cui è cambiato lo stato

statusMessage	String	Si	Messaggio di rifiuto della richiesta
---------------	--------	----	--------------------------------------

4.3. MODULI VERTICALI

4.3.1. SIT

Il modulo verticale SIT è dedicato alla fruizione dei Servizi Innovativi Territoriali. Per i riferimenti funzionali di prodotto si rimanda al documento del Progetto Esecutivo.

Il modulo verticale SIT è sviluppato su tecnologia PHP Laravel+Vite, NGINX come WebServer e database MySQL. L'autenticazione è delegata a CAS tramite il protocollo CAS Client utilizzato dalla libreria specifica PHP. L'architettura software di prodotto è strutturata per esporre in modalità multi tenant un sito specifico per ogni Provincia aderente.

L'applicativo è quindi eseguito in singola istanza con dati segregati per singolo tenant a livello di database. Ogni tenant ha un database schema di database dedicato.

L'architettura delle informazioni prevede i seguenti database:

- sit_landlord: contiene l'anagrafica dei tenant indicante il rispettivo nome DNS e il nome del database di riferimento.
- sit_<abbreviazione tenant>_entex: contiene i dati specifici di ogni tenant

dispiegamento di un tenant prevede le seguenti operazioni:

- Creazione DNS e indirizzamento verso il Load Balancer
- Creazione schema vuoto con le grant per l'utente applicativo
- Aggiungere nella tabella "tenant" del database sit_landlord:
 - Nome del nuovo database
 - DNS del tenant
 - URL di CAS per la delega dell'autenticazione
- Eseguire sul server i seeder per il popolamento della base dati

```
php artisan tenants:artisan "migrate --  
path=database/migrations/tenant --database=tenant --seed"
```

- Estendere le viste sul database sit_landlord andando a includere il nuovo database così che Apache Superset possa recepire i dati del nuovo tenant.

4.3.2. SAPE

Per il modulo SAPE è stato creato un applicativo specifico e basato direttamente sulla componente Liferay; si veda il file relativo alla definizione delle tabelle del progetto Pi. Co.

4.3.3. SUA

Per il modulo SUA è stato creato un applicativo specifico e basato direttamente sulla componente Liferay; per la parte funzionale si faccia direttamente riferimento al documento "Analisi tecnico funzionale". A seguire si veda la sezione relativa alla definizione delle tabelle del progetto SUA.

4.3.4. SUA - Entità del database

- *Struttura dell'entità Gara*

Rappresenta la singola gara d'appalto.

Nome	Tipo	Obbligatorio	Note
garaId	long	sì	Chiave primaria
companyId	long	sì	Audit field
groupId	long	sì	Audit field
userId	long	sì	Audit field
userName	String	sì	Audit field

createDate	Date	sì	Audit field
modifiedDate	Date	sì	Audit field
authorityId	long	sì	Identificativo dell'ente richiedente <i>principale</i> Dato replicato anche nell'entità GaraEnte
provinceCode	String	sì	Codice univoco della provincia Dato ridondato (si ricava da Authority)
suaAuthorizedProvinceId	long	sì	Identificativo della SUA appaltante Le SUA appaltante sono tutte le province autorizzate in cui <code>sua=true</code>
suaUserId	long	no	Identificativo dell'utente della SUA che prende in carico la gara

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

suaTakeChargeDate	Date	no	Data di presa in carico della SUA
rupUserId	long	no	Identificativo dell'utente che rappresenta il RUP
rupFullName	String	no	Nome completo dell'utente che rappresenta il RUP Dato ridondato usato esternamente da OTRS per i KPI
campagnaId	long	no	Identificativo della campagna da cui ha avuto origine la gara (e da cui sono stati importati i fabbisogni)

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

tipologiaRichiesta	int	no	<p>Tipologia della richiesta di supporto</p> <p>Può assumere i valori:</p> <ul style="list-style-type: none"> • Richiesta di supporto per Gara su Delega; • Richiesta di supporto per redazione atti di Gara
oggettoContratto	int	sì	<p>Oggetto del contratto</p> <p>Può assumere i valori:</p> <ul style="list-style-type: none"> • Forniture (default) • Lavori • Servizi
oggettoGara	String 256	sì	Oggetto della gara
descrizione	String 2000	no	Descrizione della gara
importoTotaleFabbisogni	double	sì	<p>Valore complessivo dei fabbisogni</p> <p>Si tratta della somma dei campi <code>GaraFabbisogno.prezzoTotale</code></p>
importoTotale	double	sì	<p>Importo totale della gara</p> <p>Si tratta della somma dei campi <code>GaraLotto.prezzoTotale</code></p>
modalitaSceltaContraenteId	long	sì	Modalità di scelta del contraente

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

settoreAttivita	String 1	si	Settore di attività della SA Può assumere i valori: <ul style="list-style-type: none"> • Ordinario (O) (default) • Speciale (S)
modalitaIndizioneId	long	no	Modalità di indizione

			Valorizzata solamente se settoreAttivita = S
modalitaRealizzazioneId	long	no	Modalità di realizzazione
strumentoSvolgimentoProcedura Id	long	no	Strumento per lo svolgimento della procedura
condizioneNegozziataId	long	no	Condizione che giustifica il ricorso alla procedura negoziata senza previa pubblicazione di un bando oppure senza previa indizione di una gara Valorizzata solamente se la modalità di scelta del contraente è "Procedura negoziata senza previa pubblicazione"
urgente	boolean	no	Esecuzione in somma urgenza
motivazioneUrgenzaId	long	no	Motivo dell'urgenza
link	String	no	Link della pagina di pubblicazione del bando nel sito della provincia
opzioneCollegamentoId	long	no	Opzione di collegamento dell'appalto

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

codiceCIG	String	no	Codice CIG
codiceANAC	String	no	ID ANAC
codiceMIT	String	no	ID MIT
usaCodiceCUP	boolean	no	Determina se usare il codice CUP
codiceCUP	String 15	no	Codice CUP
numeroProtocolloInterno	String	no	Numero protocollo interno

ambitoPNRR	boolean	no	Determina se la procedura è in ambito PNRR
categoriaMerceologicaId	long	sì	Identificativo della categoria merceologica Valore di default 999
motivazioneCigId	long	no	Motivazione richiesta CIG Visualizzata e valorizzata solo se <code>categoriaMerceologicaId < 999</code>
cpvPrevalenteId	long	sì	CPV prevalente
cpvSecondarioId	long	no	CPV secondario

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

dataScadenzaPresentazioneOfferte	Date	no	Data di scadenza per la presentazione delle offerte In teoria la gara dovrebbe un campo "Data di scadenza per la presentazione delle offerte"; tuttavia questo campo non è presente nelle bozze, quindi non ho idea di che valore abbia. Però sembra che serva per visualizzare un timer a video e gestire il cambio di stato; pertanto mi invento che sono 2 mesi dalla data di creazione della gara.
durataConvenzione	int	no	Durata della convenzione o accordo quadro
durataConvenzioneMisura	int	no	Unità di misura della durata della convenzione o accordo quadro
dataAvvioAppalto	Date	no	Data avvio appalto / concessione
durataAffidamento	int	no	Durata dell'affidamento
durataAffidamentoMisura	int	no	Unità di misura della durata dell'affidamento Può valere: giorni (1), mesi (2), anni (3)
luogoEsecuzioneIstatCityId	long	no	Codice del luogo di esecuzione del contratto (ISTAT) Rappresenta il comune del luogo di esecuzione

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

luogoEsecuzioneNuts	String	no	Codice del luogo di esecuzione del contratto (NUTS) Rappresenta la regione o la provincia del luogo di esecuzione del contratto, con la codifica europea NUTS
ripetizioniPreviste	boolean	no	L'appalto prevede ripetizioni o altre opzioni?
durataRinnovi	int	no	Durata dei rinnovi e delle opzioni
durataRinnoviMisura	int	no	Unità di misura della durata dei rinnovi e delle opzioni
agisceSoggettoTerzo	boolean	no	La stazione appaltante agisce per conto di altro soggetto?
funzioneDelegataId	long	no	Funzione delegata
codiceFiscaleSoggettoTerzo	String	no	Codice fiscale soggetto per conto del quale agisce la S.A. (in caso di singolo soggetto)
lavoroPrevisto	boolean	no	Il lavoro o l'acquisto di bene o servizio è stato previsto all'interno della programmazione

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

<code>primaAnnualita</code>	int	no	<p>Prima annualità dell'ultimo programma nel quale è stato inserito l'intervento o acquisto</p> <p>Formato anno (può essere inserito qualsiasi anno tra i tre precedenti a quello attuale, quello attuale e fino a 10 anni nel futuro)</p> <p>Visualizzata e valorizzata solo se <code>lavoroPrevisto = true</code></p>
<code>codiceUnicoIntervento</code>	String 25	no	<p>CUI programma triennale lavori pubblici o programma biennale forniture e servizi</p> <p>Visualizzata e valorizzata solo se <code>lavoroPrevisto = true</code></p>
<code>dataConclusionePrestazioni</code>	Date	no	Data di conclusione delle prestazioni contrattuali
<code>cancelMessage</code>	String	no	Motivo per l'annullamento della richiesta
<code>rejectMessage</code>	String	no	Motivo per il rifiuto della richiesta
<code>withdrawMessage</code>	String	no	Motivazione per il ritiro della richiesta
<code>status</code>	int	sì	Status field
<code>statusByUserId</code>	long	sì	Status field
<code>statusByUsername</code>	String	sì	Status field

statusDate	Date	sì	Status field
------------	------	----	--------------

- **Struttura dell'entità GaraReferente**

Rappresenta il singolo referente di una gara d'appalto.

Nome	Tipo	Obbligatorio	Note
garaReferente Id	long	sì	Chiave primaria
companyId	long	sì	Audit field
groupId	long	sì	Audit field
userId	long	sì	Audit field
userName	String	sì	Audit field
createDate	Date	sì	Audit field
modifiedDate	Date	sì	Audit field
garaId	long	sì	Identificativo della gara

authorityUser Id	long	sì	Identificativo dell'utente referente per l'ente richiedente Si sarebbe dovuto chiamare semplicemente userId ma è stato differenziato dal campo di audit
primary	boolean	no	Determina se il referente è quello principale per la gara Il referente principale non può essere rimosso dalla gara

Indici univoci

Campi	Descrizione
groupId, authorityUserId garaId,	Lo stesso utente non può comparire più di una volta nella stessa gara

- **Struttura dell'entità GaraFabbisogno**

Rappresenta il singolo fabbisogno della gara.

Nome	Tipo	Obbligatorio	Note
garaFabbisogno Id	long	sì	Chiave primaria
companyId	long	sì	Audit field
groupId	long	sì	Audit field
userId	long	sì	Audit field

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
 nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

userName	String	sì	Audit field
createDate	Date	sì	Audit field
modifiedDate	Date	sì	Audit field
garaId	long	sì	Identificativo della gara
garaLottoId	int	sì	Identificativo del lotto
authorityId	long	sì	Identificativo dell'ente
nome	String 50	sì	Nome del fabbisogno
descrizione	String 250	no	Descrizione del fabbisogno
cpvPrevalenteId	long	no	Codice CPV prevalente
unitaMisura	String 50	sì	Unità di misura
quantita	double	sì	Quantità

prezzoUnitario	double	sì	Prezzo unitario
prezzoTotale	double	sì	Totale di riga Campo calcolato in automatico

- *Struttura dell'entità* **GaraLotto**

Rappresenta il singolo lotto di cui fanno parte i fabbisogni della gara.

Nome	Tipo	Obbligatorio	Note
garaLottoId	long	sì	Chiave primaria
companyId	long	sì	Audit field
groupId	long	sì	Audit field
userId	long	sì	Audit field
userName	String	sì	Audit field
createDate	Date	sì	Audit field
modifiedDate	Date	sì	Audit field
garaId	long	sì	Identificativo della gara

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

lotto	int	sì	Numero del lotto
codiceCIG	String	no	Codice CIG
prezzoTotale	double	sì	Importo totale del lotto Campo calcolato in automatico
prezzoOpzioni	double	no	Di cui per opzioni
prezzoSicurezza	double	no	Di cui per attuazione della sicurezza
cpvPrevalenteId	long	no	Codice CPV prevalente
durata	int	no	Durata contratto (in giorni)
luogoEsecuzioneIstatCityId	long	no	Codice del luogo di esecuzione del contratto (ISTAT)

Indici univoci

Campi	Descrizione
groupId, lotto, garaId,	Non possono esistere 2 lotti con lo stesso numero in una gara

• **Struttura dell'entità GaraDocumento**

Rappresenta il singolo documento associato alla gara.

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

Nome	Tipo	Obbligatorio	Note
garaDocumentoId	long	sì	Chiave primaria
companyId	long	sì	Audit field
groupId	long	sì	Audit field
userId	long	sì	Audit field
userName	String	sì	Audit field
createDate	Date	sì	Audit field
modifiedDate	Date	sì	Audit field
garaId	long	sì	Identificativo della gara
garaLottoId	long	no	Lotto di appartenenza Se vale 0 significa che il documento è relativo a tutti i lotti
authorityId	long	no	Ente di appartenenza dell'utente che ha caricato il documento
type	int	no	Tipo di documento I valori ammissibili sono censiti nella classe <code>GaraDocumentoType</code>
descrizione	String 100	sì	Descrizione del documento
fileEntryId	long	no	Identificativo del file

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

fileSharingURL	String 255	no	Link file sharing
status	int	sì	Status field Può assumere i valori: APPROVED e ARCHIVED
statusById	long	sì	Status field
statusByUserName	String	sì	Status field
statusDate	Date	sì	Status field

• *Struttura dell'entità* **CategoriaMerceologica** Rappresenta la singola categoria merceologica.

Nome	Tipo	Obbligatorio	Note
categoriaMerceologicaId	long	sì	Chiave primaria
companyId	long	sì	Audit field
groupId	long	sì	Audit field I record sono globali per tutti i siti e definiti a livello di company
userId	long	sì	Audit field

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
 nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

userName	String	si	Audit field
createDate	Date	si	Audit field
modifiedDate	Date	si	Audit field
code	int	si	Codice
description	String 100	si	Descrizione

Indici univoci

Campi	Descrizione
companyId, code	Non possono esistere 2 categorie con lo stesso codice
companyId, description	Non possono esistere 2 categorie con lo stessa descrizione

- *Struttura dell'entità*

CondizioneNegozziata

Rappresenta la singola condizione che giustifica il ricorso alla procedura negoziata senza previa pubblicazione di un bando oppure senza previa indizione di una gara.

Nome	Tipo	Obbligatorio	Note
condizioneNegozziataId	long	sì	Chiave primaria
companyId	long	sì	Audit field
groupId	long	sì	Audit field I record sono globali per tutti i siti e definiti a livello di company
userId	long	sì	Audit field
userName	String	sì	Audit field
createDate	Date	sì	Audit field
modifiedDate	Date	sì	Audit field
code	int	sì	Codice
description	String 255	sì	Descrizione

- *Struttura dell'entità*

Indici univoci

Campi	Descrizione
companyId, code	Non possono esistere 2 condizioni con lo stesso codice
companyId, description	Non possono esistere 2 condizioni con lo stessa descrizione

CPV

Rappresenta il singolo CPV (Common Procurement Vocabulary) ossia la classificazione della gara.

Non è chiaro se si tratti di un vocabolario, quindi è stata modellata con una tabella custom.

Nome	Tipo	Obbligatorio	Note
cpvId	long	sì	Chiave primaria
companyId	long	sì	Audit field
groupId	long	sì	Audit field I record sono globali per tutti i siti e definiti a livello di company
userId	long	sì	Audit field
userName	String	sì	Audit field

• *Struttura dell'entità*

<code>createDate</code>	Date	sì	Audit field
<code>modifiedDate</code>	Date	sì	Audit field
<code>parentCpvId</code>	long	no	CPV padre
<code>code</code>	int	sì	Codice
<code>description</code>	String 255	sì	Descrizione

Indici univoci

Campi	Descrizione
<code>companyId, code</code>	Non possono esistere 2 categorie con lo stesso codice
<code>companyId, description</code>	Non possono esistere 2 categorie con lo stessa descrizione

FunzioneDelegata Rappresenta

la singola funzione delegata.

Nome	Tipo	Obbligatorio	Note
<code>funzioneDelegataId</code>	long	sì	Chiave primaria
<code>companyId</code>	long	sì	Audit field

• *Struttura dell'entità*

groupId	long	sì	Audit field I record sono globali per tutti i siti e definiti a livello di company
userId	long	sì	Audit field
userName	String	sì	Audit field
createDate	Date	sì	Audit field
modifiedDate	Date	sì	Audit field
code	int	sì	Codice
description	String 255	sì	Descrizione

Indici univoci

Campi	Descrizione
companyId, code	Non possono esistere 2 modalità con lo stesso codice
companyId, description	Non possono esistere 2 modalità con lo stessa descrizione

ModalitàIndizione

• *Struttura dell'entità*

Rappresenta la singola modalità di indizione, selezionabile solamente se il campo
 Gara.settoreAttivita = S.

Nome	Tipo	Obbligatorio	Note
modalitaIndizione Id	long	sì	Chiave primaria
companyId	long	sì	Audit field
groupId	long	sì	Audit field I record sono globali per tutti i siti e definiti a livello di company
userId	long	sì	Audit field
userName	String	sì	Audit field
createDate	Date	sì	Audit field
modifiedDate	Date	sì	Audit field
code	int	sì	Codice
description	String 255	sì	Descrizione

Indici univoci

Campi	Descrizione

• *Struttura dell'entità*

<code>companyId, code</code>	Non possono esistere 2 modalità con lo stesso codice
<code>companyId, description</code>	Non possono esistere 2 modalità con lo stessa descrizione

ModalitaRealizzazione Rappresenta

la singola modalità di realizzazione.

Nome	Tipo	Obbligatorio	Note
<code>modalitaRealizzazione Id</code>	long	sì	Chiave primaria
<code>companyId</code>	long	sì	Audit field
<code>groupId</code>	long	sì	Audit field I record sono globali per tutti i siti e definiti a livello di company
<code>userId</code>	long	sì	Audit field
<code>userName</code>	String	sì	Audit field
<code>createDate</code>	Date	sì	Audit field

• *Struttura dell'entità*

modifiedDate	Date	sì	Audit field
code	int	sì	Codice
description	String 255	sì	Descrizione

Indici univoci

Campi	Descrizione
companyId, code	Non possono esistere 2 modalità con lo stesso codice
companyId, description	Non possono esistere 2 modalità con lo stessa descrizione

ModalitaSceltaContraente

Rappresenta la singola modalità di scelta del contraente, selezionabile solamente quando la gara raggiunge lo stato *Definizione tipologia procedura*.

Nome	Tipo	Obbligatorio	Note
modalitaSceltaContraenteId	long	sì	Chiave primaria
companyId	long	sì	Audit field
groupId	long	sì	Audit field I record sono globali per tutti i siti e definiti a livello di company

• *Struttura dell'entità*

<code>userId</code>	long	si	Audit field
<code>userName</code>	String	si	Audit field
<code>createDate</code>	Date	si	Audit field
<code>modifiedDate</code>	Date	si	Audit field
<code>code</code>	int	si	Codice
<code>description</code>	String 255	si	Descrizione
<code>invitation</code>	boolean	no	Determina se la modalità di scelta è su invito oppure aperta

Indici univoci

Campi	Descrizione
<code>companyId, code</code>	Non possono esistere 2 modalità con lo stesso codice
<code>companyId, description</code>	Non possono esistere 2 modalità con lo stessa descrizione

MotivazioneCIG

Rappresenta la singola motivazione della richiesta CIG.

• *Struttura dell'entità*

Nome	Tipo	Obbligatorio	Note
motivazioneCig Id	long	sì	Chiave primaria
companyId	long	sì	Audit field
groupId	long	sì	Audit field I record sono globali per tutti i siti e definiti a livello di company
userId	long	sì	Audit field
userName	String	sì	Audit field
createDate	Date	sì	Audit field
modifiedDate	Date	sì	Audit field
code	int	sì	Codice
description	String 100	sì	Descrizione

Indici univoci

Campi	Descrizione

• *Struttura dell'entità*

companyId, code	Non possono esistere 2 motivazioni con lo stesso codice
companyId, description	Non possono esistere 2 motivazioni con lo stessa descrizione

MotivazioneUrgenza

Rappresenta il singolo motivo per la richiesta dei lavori in urgenza.

Nome	Tipo	Obbligatorio	Note
motivazioneUrgenza Id	long	sì	Chiave primaria
companyId	long	sì	Audit field
groupId	long	sì	Audit field I record sono globali per tutti i siti e definiti a livello di company
userId	long	sì	Audit field
userName	String	sì	Audit field
createDate	Date	sì	Audit field
modifiedDate	Date	sì	Audit field

• *Struttura dell'entità*

code	int	sì	Codice
description	String 255	sì	Descrizione

Indici univoci

Campi	Descrizione
companyId, code	Non possono esistere 2 motivazioni con lo stesso codice
companyId, description	Non possono esistere 2 motivazioni con lo stessa descrizione

OpzioneCollegamento

Rappresenta la singola opzione di collegamento dell'appalto.

Nome	Tipo	Obbligatorio	Note
opzioneCollegamentoId	long	sì	Chiave primaria
companyId	long	sì	Audit field
groupId	long	sì	Audit field I record sono globali per tutti i siti e definiti a livello di company
userId	long	sì	Audit field

• *Struttura dell'entità*

<code>userName</code>	String	sì	Audit field
<code>createDate</code>	Date	sì	Audit field
<code>modifiedDate</code>	Date	sì	Audit field
<code>code</code>	int	sì	Codice
<code>description</code>	String 255	sì	Descrizione

Indici univoci

Campi	Descrizione
<code>companyId, code</code>	Non possono esistere 2 opzioni con lo stesso codice
<code>companyId, description</code>	Non possono esistere 2 opzioni con lo stessa descrizione

StrumentoSvolgimentoProcedura

Rappresenta il singolo strumento per lo svolgimento della procedura.

Nome	Tipo	Obbligatorio	Note
<code>strumentoSvolgimentoProceduraId</code>	long	sì	Chiave primaria

• *Struttura dell'entità*

companyId	long	sì	Audit field
groupId	long	sì	Audit field I record sono globali per tutti i siti e definiti a livello di company
userId	long	sì	Audit field
userName	String	sì	Audit field
createDate	Date	sì	Audit field
modifiedDate	Date	sì	Audit field
code	int	sì	Codice
description	String 255	sì	Descrizione

Indici univoci

Campi	Descrizione
companyId, code	Non possono esistere 2 strumenti con lo stesso codice
companyId, description	Non possono esistere 2 strumenti con lo stessa descrizione

- *Struttura dell'entità*

5. MODULO FORMAZIONE

5.1. INTRODUZIONE

In risposta alle specifiche esigenze di **UPI** riguardanti la presenza di un modulo formativo, è stata individuata in **IOMAD** la soluzione di learning management system ideale per gestire e strutturare la formazione destinata ai diversi gruppi di destinatari.

IOMAD, costruito su Moodle 4.1.4, è stato scelto per la sua **capacità di gestire più istanze di formazione** in modo centralizzato, permettendo una gestione efficiente delle diverse esigenze formative all'interno della piattaforma.

Grazie alla possibilità di creare istanze indipendenti per ciascun gruppo destinatario e alla gestione centralizzata tramite IOMAD, UPI potrà garantire una **formazione scalabile, sicura e facilmente monitorabile**. Inoltre, l'integrazione dei contenuti formativi con gli altri moduli di **Community** e **Helpdesk** della piattaforma PI.CO potrà arricchire l'esperienza di apprendimento e favorire una formazione continua e collaborativa tra i vari enti aderenti. IOMAD si rivela così la soluzione più adatta per supportare UPI nell'implementazione di un piano formativo efficace e mirato per l'intero ecosistema della piattaforma PI.CO.

5.2. LA SOLUZIONE

IOMAD è una soluzione open-source avanzata che si costruisce sulla robustezza di Moodle, offrendo un ambiente di apprendimento online multi-tenant. Ciò significa che consente di gestire più istanze di Moodle in modo completamente indipendente all'interno di una singola installazione. Questa caratteristica risulta particolarmente vantaggiosa per organizzazioni complesse che desiderano amministrare vari siti Moodle, ciascuno con configurazioni, utenti e contenuti specifici, mantenendo però un'infrastruttura centrale.

Il sistema consente una gestione centralizzata e altamente scalabile, semplificando le operazioni quotidiane, dalla gestione degli utenti alla personalizzazione dei corsi, e garantendo nel contempo la massima sicurezza e conformità alle normative. Inoltre, la capacità di IOMAD di supportare un ampio numero di utenti e corsi rende la piattaforma ideale per le organizzazioni che cercano di ottimizzare le risorse e di ridurre i costi, offrendo al contempo un'esperienza di apprendimento online diversificata e personalizzata per ogni gruppo di utenti.

Una delle principali forze di IOMAD è la possibilità di personalizzare l'esperienza di apprendimento per ogni tenant, consentendo ad esempio la creazione di percorsi formativi su misura, la gestione delle certificazioni. IOMAD si distingue inoltre per il suo supporto alle funzionalità avanzate di reporting e analisi, che consentono alle organizzazioni di monitorare in tempo reale i progressi degli utenti e l'efficacia dei corsi.

Grazie alla sua architettura modulare, IOMAD offre anche un'elevata compatibilità con plug-in esterni e personalizzazioni, permettendo alle organizzazioni di evolvere e adattare la piattaforma alle proprie esigenze in modo agile e senza compromettere le performance.

5.3. INFRASTRUTTURA NECESSARIA

5.3.1. Server

L'infrastruttura di base di IOMAD richiede un server dedicato in grado di supportare un'installazione di Moodle multi-tenant. Le risorse del server devono essere proporzionate al numero di istanze Moodle che si intende gestire, al numero di utenti e alla quantità di dati che verranno elaborati. È fondamentale avere un processore veloce e multi-core per gestire le richieste contemporanee, una RAM sufficiente per evitare rallentamenti e blocchi, e spazio su disco adeguato per contenere i dati degli utenti, i corsi e le risorse multimediali. Inoltre, la banda di rete deve essere abbastanza larga

per supportare il traffico web proveniente dagli utenti che accedono alle piattaforme, evitando sovraccarichi che potrebbero causare disservizi.

Le prestazioni di IOMAD possono essere ottimizzate tramite diverse strategie, tra cui la corretta dimensionamento dell'hardware (ad esempio, utilizzando server con maggiore potenza di calcolo e maggiore memoria RAM) e l'adozione di tecniche di caching per ridurre i tempi di caricamento delle pagine e il carico sul database. Inoltre, l'architettura modulare permette di disabilitare moduli non necessari, migliorando le prestazioni generali. Infine, l'adozione di tecniche di **bilanciamento del carico** e l'utilizzo di **CDN (Content Delivery Networks)** possono ottimizzare ulteriormente l'esperienza utente, garantendo accesso rapido anche in scenari con un gran numero di utenti simultanei.

Ricordiamo che per il progetto UPI si utilizza una infrastruttura Azure e quindi con la possibilità di scalare rapidamente in caso di necessità per spazio storage e potenza di elaborazione.

5.3.2. Sistema operativo

IOMAD, come Moodle, richiede un sistema operativo stabile e affidabile. Le due opzioni più comuni sono **CentOS** e **Ubuntu**, che sono distribuzioni di Linux ampiamente supportate e largamente utilizzate per ambienti di produzione. CentOS è noto per la sua stabilità e per la compatibilità con software enterprise.

5.3.3. Database

IOMAD si basa su un database relazionale per archiviare i dati delle diverse istanze Moodle, inclusi utenti, corsi, risultati, e risorse. Le due opzioni più comuni per il database sono MySQL, PostgreSQL, Maria Db.

Per l'istanza Pi.Co è stato utilizzato database **Maria Db 5.5.5**

5.3.4. Web server

Per servire le pagine web di Moodle e IOMAD, è necessario un web server come **Apache** o **Nginx**. Apache è il web server più tradizionale e offre una configurazione completa per gestire richieste

HTTP e HTTPS, oltre a numerosi moduli che possono essere utilizzati per estendere le funzionalità.

5.3.5. PHP

PHP è il linguaggio di programmazione su cui è costruito Moodle (e quindi IOMAD). È fondamentale che il server utilizzi una versione di **PHP compatibile con la versione di Moodle che si sta utilizzando**. Attualmente, la versione di Moodle è la 4.1.4 e supporta versioni di PHP dalla 7.9 in su. Attualmente è installata la versione PHP 8.1.2.1

5.3.6. Software aggiuntivo

IOMAD richiede alcuni strumenti aggiuntivi per gestire correttamente le dipendenze e il ciclo di vita del software. **Composer** è un gestore di pacchetti per PHP che permette di installare e aggiornare le librerie di terze parti necessarie per il funzionamento di Moodle e IOMAD. **npm** (Node Package Manager) è necessario per gestire le dipendenze JavaScript, che possono includere librerie e strumenti per migliorare l'interfaccia utente di Moodle. Entrambi questi strumenti sono essenziali per la gestione delle librerie e per garantire che il sistema resti aggiornato con le ultime versioni delle dipendenze, evitando conflitti o malfunzionamenti.

5.4. FUNZIONALITÀ PRINCIPALI

5.4.1. Multi-tenancy

IOMAD sfrutta il concetto di *multi-tenancy* per permettere la gestione di più istanze di Moodle in un'unica installazione. Ogni istanza agisce in modo indipendente, il che significa che ogni "tenant" (ad esempio, UPI, Province) può avere la propria configurazione, utenti, corsi e contenuti. Gli amministratori possono creare nuove istanze facilmente, assegnare risorse e gestire gli utenti senza interferenze tra un'istanza e l'altra. Questo consente di soddisfare esigenze molto diverse all'interno della stessa infrastruttura, rendendo IOMAD ideale per ambienti multi-organizzativi. Nel caso del portale Pi-co, esiste una struttura gerarchica per cui UPI come istanza può facilmente pubblicare corsi aperti a tutte le istanze delle Province.

Le province invece hanno il controllo esclusivo della propria istanza e quindi possono gestire il proprio catalogo formativo.

5.4.2. Gestione centralizzata

Una delle caratteristiche distintive di IOMAD è la gestione centralizzata tramite un'unica interfaccia amministrativa "IOMAD Dashboard". Gli amministratori di sistema possono accedere a tutte le istanze Moodle dalla stessa piattaforma, consentendo di configurare, monitorare e gestire i vari tenant senza dover accedere a ciascuna istanza separatamente. Questo rende la gestione più efficiente, riducendo il tempo necessario per attività come l'aggiornamento delle configurazioni, l'assegnazione degli utenti e la supervisione delle attività di apprendimento.

5.4.3. Condivisione delle risorse

Con IOMAD è possibile condividere risorse comuni tra le diverse istanze, facilitando la gestione centralizzata di contenuti e strumenti. Ad esempio, gli amministratori possono caricare blocchi, temi o plugin che possono essere utilizzati da più istanze, riducendo la duplicazione dei contenuti e ottimizzando l'infrastruttura. Questo è particolarmente utile quando si desidera applicare aggiornamenti o modifiche a livello globale senza dover intervenire singolarmente su ogni istanza. Inoltre, la condivisione di risorse consente alle organizzazioni di standardizzare l'esperienza utente tra le diverse istanze, garantendo coerenza nell'interfaccia e nelle funzionalità.

5.4.4. Sicurezza

La sicurezza è un aspetto fondamentale di IOMAD. Ogni istanza può disporre di un proprio sistema di autenticazione e autorizzazione, che può essere personalizzato per soddisfare le specifiche esigenze di ciascun tenant. IOMAD supporta vari metodi di autenticazione, inclusi login tramite **Single Sign-On (SSO)**, **LDAP**, **autenticazione via social media** e altro ancora. Inoltre, le politiche di autorizzazione possono essere configurate a livello di singola istanza o a livello globale per garantire che solo gli utenti autorizzati accedano a determinati contenuti o funzioni. La protezione dei dati è assicurata da misure avanzate di crittografia e da una gestione rigorosa dei permessi.

5.5. CARATTERISTICHE TECNICHE

5.5.1. Architettura modulare

IOMAD si basa su un'architettura modulare che offre una grande flessibilità e personalizzazione. La modularità permette agli sviluppatori di aggiungere o rimuovere facilmente moduli in base alle esigenze specifiche dell'organizzazione. Ad esempio, è possibile integrare nuovi strumenti di valutazione, reportistica avanzata, funzionalità di gamification o sistemi di gestione della formazione esterni. Ogni modulo può essere sviluppato e aggiornato separatamente, riducendo i tempi di sviluppo e di manutenzione, e permettendo agli amministratori di gestire solo ciò che è necessario per il loro ambiente di apprendimento.

5.5.2. API REST

IOMAD offre un'API RESTful che consente di integrarlo facilmente con altri sistemi aziendali, applicazioni o piattaforme esterne. Le API (Application Programming Interfaces) sono essenziali per garantire che IOMAD possa comunicare e scambiarsi dati con altri software, come sistemi di gestione delle risorse umane (HRM), software di gestione aziendale (ERP), piattaforme di videoconferenza o sistemi di e-commerce. L'adozione di un'architettura RESTful rende l'integrazione più semplice e scalabile, poiché le API REST sono leggere, facili da implementare e utilizzare in modo standardizzato. Grazie a questa API, IOMAD può diventare una parte integrante di un ecosistema più ampio, migliorando l'efficienza operativa e favorendo l'automazione dei flussi di lavoro.

5.5.3. Integrazione con Liferay Portal

IOMAD supporta l'integrazione con vari sistemi di autenticazione esterni, come **LDAP** (Lightweight Directory Access Protocol) e **SAML** (Security Assertion Markup Language), così come altri protocolli di autenticazione. Questo significa che le organizzazioni possono centralizzare la gestione degli utenti e garantire una maggiore sicurezza attraverso sistemi di autenticazione consolidati. Per il progetto UPI l'accesso a IOMAD avviene attraverso la registrazione e l'autenticazione sul portale PI.CO. La registrazione dell'utente attraverso il sistema CAS garantisce parallelamente la registrazione via LDAP su Liferay attraverso una serie di parametri condivisi ed una chiave primaria basata su codice fiscale con prefisso TINIT.

IOMAD poi utilizza le funzionalità native ed i plugin disponibili per la relativa importazione delle utenze a livello di istanza UPI e delle singole Province sulla base dei dati di profilazione iniziale dell'utente.

5.5.4. Integrazioni

- **Integrazione con strumenti di videoconferenza:** IOMAD si può integrare facilmente con piattaforme di **videoconferenza** come **Zoom**, **BigBlueButton** o **Microsoft Teams**. Questo consente di gestire sessioni di formazione in tempo reale direttamente all'interno delle istanze di Moodle, arricchendo l'esperienza di apprendimento online con sessioni di gruppo interattive. Attualmente sulla istanza UPI è disponibile per un numero limitato di utenti, il servizio di videoconferenza BigBlueButton che può essere comodamente attivato con e risorsa di un corso.
- **Integrazione con strumenti di business intelligence:** Utilizzando plugin o soluzioni esterne, IOMAD può integrarsi con strumenti di **business intelligence (BI)**, per analizzare i dati relativi all'apprendimento e alle performance degli utenti, generando report avanzati e dashboards personalizzate. Per il riavvio del progetto diversamente per quello che succede sui moduli verticali SIT, SUA e SAPE non è stato richiesto l'implementazione di dashboard personalizzate con i dati forniti dal LMS.

5.6. GESTIONE DELLE ISTANZE SU IOMAD

5.6.1. Creazione di nuove istanze

Il processo di creazione di nuove istanze in IOMAD è altamente semplificato per ridurre i tempi e gli errori. Ogni nuova istanza può essere configurata in modo indipendente, garantendo che ogni "tenant" (ad esempio, un dipartimento o un cliente) abbia le proprie specifiche preferenze, come la lingua, la struttura dei corsi, le modalità di accesso e i permessi. Questo processo non richiede competenze tecniche avanzate, rendendolo accessibile anche a chi non ha un background IT, ma desidera comunque gestire autonomamente la propria piattaforma di apprendimento online. Grazie a configurazioni personalizzabili, le organizzazioni possono adattare ogni istanza alle proprie necessità senza dover intervenire pesantemente sulla struttura di base.

5.6.2. Clonazione di istanze

La clonazione delle istanze è una funzionalità chiave per semplificare e velocizzare la creazione di nuove piattaforme all'interno di un'infrastruttura esistente. Quando si ha una configurazione ben definita e consolidata, è possibile clonarla e creare una nuova istanza in pochi passaggi, risparmiando tempo e riducendo gli errori dovuti alla configurazione manuale. Questa funzionalità è particolarmente utile nel contesto UPI in cui si vogliono replicare rapidamente ambienti di apprendimento simili per le rispettive provincie. La clonazione permette di duplicare rapidamente l'intera configurazione, i corsi, i temi e i permessi di un'istanza preesistente, mantenendo la coerenza nelle operazioni e facilitando la gestione complessiva.

5.6.3. Aggiornamenti

Con IOMAD, la gestione degli aggiornamenti è centralizzata, il che significa che gli amministratori possono eseguire gli aggiornamenti di Moodle e IOMAD tramite un'unica interfaccia di gestione, evitando il bisogno di aggiornare ogni istanza singolarmente. Questo approccio semplifica la manutenzione e migliora l'efficienza, poiché gli aggiornamenti possono essere applicati contemporaneamente a tutte le istanze, garantendo che ogni piattaforma resti sempre aggiornata con le ultime funzionalità e patch di sicurezza. L'approccio centralizzato riduce i rischi associati a versioni obsolete o non allineate, e consente di eseguire controlli di compatibilità prima di applicare gli aggiornamenti a tutte le istanze, minimizzando il rischio di interruzioni del servizio.

5.6.4. Backup e ripristino

La possibilità di creare backup completi delle istanze e di ripristinarle facilmente è essenziale per garantire la sicurezza e la continuità operativa delle piattaforme Moodle gestite tramite IOMAD. I backup possono essere configurati in modo da essere eseguiti automaticamente a intervalli regolari, e possono includere dati degli utenti, configurazioni, contenuti dei corsi e risorse condivise. In caso di necessità, come un errore tecnico o un attacco informatico, le istanze possono essere ripristinate in tempi rapidi, riducendo al minimo i tempi di inattività. IOMAD offre anche la possibilità di eseguire backup differenziati per ogni istanza, permettendo una gestione mirata delle risorse e garantendo la protezione dei dati in scenari complessi. Il sistema di backup e ripristino aiuta a prevenire la perdita di dati e a mantenere un elevato livello di affidabilità operativa.

Si rimanda alle specifiche policy di backup della infrastruttura UPI su Azure.

5.7. AMMINISTRAZIONE DI IOMAD

Iomad è una versione customizzata di Moodle e pertanto le logiche di amministrazione sono le medesime di Moodle 4.0 in poi e gestibili per ogni istanza attraverso "Amministrazione del sito" e relative funzionalità quali: Generale, utenti, corsi, valutazioni., Plugin, aspetto., Etc... Iomad inoltre dispone di una Iomad Dashboard da cui gestire in modo immediato i corsi, le utenze, report ed altre funzionalità principali.

Riportiamo nello specifico il passaggio per la creazione di un corso sulla istanza UPI ma che vale per qualsiasi istanza presente.

5.7.1. Creazione corsi

Cliccando entrando nella sezione corsi, cliccando su crea corsi è possibile procedere con la creazione di un corso punto

Per creare un corso è necessario decidere il titolo del corso e un titolo abbreviato che vengono richiesti perché obbligatori e una breve introduzione il cui inserimento è però facoltativo.

Una volta compilati i campi richiesti scorriamo verso il basso e clicchiamo **Crea e vai al corso**. In questo modo si verrà ridiretti alla pagina di iscrizione del corso. Per poter accedere al corso e iniziare a caricare i contenuti occorre cliccare su **Iscrivimi**.

Per cominciare a caricare contenuti occorre abilitare la **Modalità modifica** con l'interruttore in alto a destra. Una volta attivata la modalità di modifica è possibile inserire un'introduzione al corso accedendo alla sezione di inserimento risorsa.

Dalla pagina di selezione sarà possibile scegliere la risorsa utile al contenuto che si vuole caricare che può essere video, pacchetto SCORM, un contenuto di testo e molto altro.

Una volta aggiunta la risorsa cliccando **Salva e torna al corso** si ritorna alla pagina iniziale del corso.

5.7.2. Abilitazione corso agli utenti

Terminate le operazioni di creazione, caricamento contenuti e configurazione del corso, occorre procedere con l'abilitazione del corso agli utenti per fare in modo che il corso possa essere disponibile per la fruizione. Per fare questo è necessario accedere alla dashboard di IOMAD e selezionare la voce **Manage IOMAD course settings** dal tab **Corsi**.

Si aprirà una schermata dove sarà possibile effettuare una ricerca del corso appena creato e popolato.

Una volta identificato il corso occorre cambiare l'impostazione della voce **Is shared** da **No** a **Open sharing** così che il corso possa essere disponibile per l'iscrizione da parte degli utenti.

6. HELPDESK (OTRS)

OTRS, acronimo di Open Ticket Request System, è nato come una soluzione open source per la gestione delle richieste di assistenza e il supporto clienti. Fin dalla sua nascita, OTRS si è contraddistinto per la sua capacità di adattarsi alle esigenze di realtà molto diverse, da piccole aziende a grandi organizzazioni internazionali. Con il tempo, OTRS è stato sviluppato e arricchito, diventando uno strumento fondamentale per la gestione dei flussi di lavoro legati al customer service, all'help desk e al supporto IT.

Funzionalità Chiave

Tra le funzionalità principali di OTRS troviamo:

- **Gestione dei Ticket:** Consente di creare, monitorare e gestire le richieste di assistenza (ticket) in maniera centralizzata.
- **Code e Servizi:** I ticket possono essere organizzati in code, servizi o dipartimenti, facilitando l'assegnazione e il trattamento in base alla priorità o alla specializzazione.
- **SLA e Tempi di Risposta:** Supporta la definizione di accordi sui livelli di servizio (Service Level Agreement), monitorando i tempi di risposta e risoluzione per garantire il rispetto degli standard aziendali.
- **Automazione e Workflow:** Permette di automatizzare azioni ripetitive e configurare flussi di lavoro personalizzati, riducendo i tempi di intervento e migliorando l'efficienza operativa.
- **Notifiche e Integrazione Email:** Invia aggiornamenti automatici via email a clienti e operatori, integrandosi con sistemi di posta elettronica per una comunicazione fluida.
- **Reportistica e Monitoraggio:** Offre strumenti di analisi e reportistica per monitorare le performance del servizio, individuare colli di bottiglia e migliorare i processi interni.
- **Portale Self-Service:** Molte implementazioni includono un portale self-service che consente agli utenti finali di aprire ticket, controllare lo stato delle richieste e consultare una knowledge base.

Architettura e Personalizzazione

OTRS è stato progettato con un'architettura modulare e scalabile, che permette di integrarlo facilmente con altri sistemi aziendali (ad es. CRM, sistemi di monitoraggio, piattaforme ERP). La gestione dei permessi, dei ruoli e dei gruppi offre un elevato grado di personalizzazione, rendendo possibile configurare il sistema in base alle specifiche esigenze organizzative e operative.

Evoluzione Verso il Modello Commerciale

Negli anni, OTRS è passato da una pura soluzione open source a un prodotto offerto anche in versione commerciale. Questa evoluzione ha portato all'introduzione di funzionalità premium, supporto dedicato e servizi aggiuntivi, rivolgendosi a realtà che necessitano di soluzioni "chiavi in mano" e contratti di assistenza a lungo termine. **Znuny: Il Fork Open Source di OTRS**

Con il progressivo spostamento di OTRS verso un modello commerciale e l'introduzione di versioni proprietarie, una parte della comunità di utenti e sviluppatori ha deciso di mantenere viva la versione open source. Da questa esigenza è nato Znuny, un fork di OTRS che mantiene intatta la filosofia originale di apertura e personalizzazione. Caratteristiche distintive di Znuny sono:

- **Filosofia Open Source:** Znuny si propone di rimanere completamente open source, consentendo a chiunque di utilizzare, modificare e ridistribuire il software senza vincoli legati a licenze commerciali.
- **Compatibilità con OTRS:** Pur mantenendo le fondamenta e la struttura di OTRS, Znuny integra miglioramenti, correzioni e nuove funzionalità sviluppate dalla community.
- **Aggiornamenti e Supporto Comunitario:** Essendo guidato da una community attiva, Znuny beneficia di aggiornamenti regolari, patch di sicurezza e nuove funzionalità che rispondono alle esigenze reali degli utenti, garantendo al contempo trasparenza e partecipazione nel processo decisionale.
- **Flessibilità e Personalizzazione:** Come la sua controparte originaria, Znuny offre una notevole flessibilità in termini di configurazione, gestione dei flussi di lavoro e integrazione con altri sistemi, risultando una scelta ideale per aziende che desiderano mantenere il controllo totale sul proprio software.

Molte organizzazioni, in particolare quelle che privilegiano soluzioni open source e necessitano di una piattaforma altamente personalizzabile senza le limitazioni o i costi legati a un modello commerciale, trovano in Znuny un'alternativa solida e affidabile. Il fork permette di mantenere vivo l'ecosistema di OTRS, garantendo agli utenti una piattaforma libera da vincoli proprietari e che si evolve grazie al contributo della comunità globale.

6.1. CREAZIONE DEI TICKET

6.1.1. Modalità di Creazione

OTRS/ZNUNY permette la creazione dei ticket attraverso diverse modalità:

- **Email:**
Le email inviate a indirizzi configurati (ad es. support@azienda.com) vengono automaticamente trasformate in ticket. Durante l'analisi dell'email, OTRS/ZNUNY estrae l'oggetto, il corpo del messaggio e gli eventuali allegati, associandoli ai campi del ticket.
- **Portale Self-Service:** Gli utenti finali possono accedere a un'interfaccia web in cui compilare un form per l'apertura di un ticket. Il form solitamente richiede:
 - Titolo/Oggetto: Breve descrizione del problema.
 - Descrizione Dettagliata: Informazioni, sintomi e contesto.
 - Servizio/Tipo di Richiesta: Selezione di una categoria predefinita (es. "Assistenza Tecnica", "Problemi di Accesso").

- Priorità: Valutazione dell'urgenza del problema.
 - Allegati: Possibilità di caricare file, screenshot o log.
- **Creazione Manuale da parte degli Agenti:** Gli agenti possono creare ticket direttamente tramite l'interfaccia amministrativa, ad esempio durante il contatto telefonico o di persona, inserendo manualmente tutte le informazioni necessarie.

6.1.2. Campi e Informazioni Gestite

Durante la creazione, il sistema gestisce numerosi campi e metadati:

- ID Ticket: Identificativo univoco generato automaticamente.
- Data/Ora di Creazione: Registrazione del momento in cui il ticket viene aperto.
- Informazioni del Richiedente: Nome, email, azienda o altri dati di contatto.
- Stato Iniziale: Generalmente "Nuovo" o "Aperto".
- Codifica del Servizio e Priorità: Utilizzate per il routing e per definire i tempi di risposta in base agli SLA (Service Level Agreements).

6.1.3. Gestione e Aggiornamento dei Ticket

Una volta creato, il ticket attraversa diverse fasi fino alla risoluzione. Ecco come si gestiscono le attività correlate:

Assegnazione e Routing

- Code (Queues):
I ticket vengono assegnati a code specifiche in base alla categoria, alla priorità o ad altre regole di routing. Le code possono essere organizzate per team o dipartimenti (es. "Supporto Tecnico", "Assistenza Clienti") e definiscono quali agenti saranno incaricati di gestire il ticket.
- Assegnazione a Agenti:
Una volta in coda, il ticket può essere:
 - Assegnato Manualmente: Un supervisore o un operatore lo assegna a un collega specifico.
 - Assegnato Automaticamente: Tramite regole configurate, il sistema distribuisce i ticket agli agenti in base al carico di lavoro, alle competenze o ad altri criteri. **Aggiornamento del Ticket**

Durante il ciclo di vita, il ticket può essere aggiornato in vari modi:

- Aggiunta di Note e Commenti:
 - Note Interne: Utili per documentare le attività e le comunicazioni interne (visibili solo agli agenti).
 - Note Esterne: Messaggi o risposte che vengono inviati al cliente, integrati nel ticket per tenere traccia delle interazioni.
- Modifica dello Stato:
Il ticket passa attraverso vari stati (es. "Nuovo", "In Lavorazione", "In Attesa", "Risolto") che riflettono il progresso nella gestione della richiesta.
- Aggiornamento delle Informazioni:
È possibile modificare la priorità, aggiornare la descrizione o aggiungere ulteriori dettagli (come allegati o link a documenti interni) per facilitare la risoluzione del problema.

Automazione e Workflow

OTRS/ZNUNY consente di definire workflow automatizzati per:

- Trigger e Condizioni: Ad esempio, se un ticket con priorità "Alta" non viene aggiornato entro 24 ore, il sistema può attivare automaticamente una regola di escalation.
- Azioni Automatiche: Inviare notifiche via email, modificare automaticamente lo stato o riassegnare il ticket a un operatore o a una coda specifica.
- Integrazione con SLA: I workflow possono essere configurati per monitorare i tempi di risposta e di risoluzione, attivando azioni correttive in caso di ritardi.

6.1.4. Risoluzione e Chiusura del Ticket

- Quando l'operatore ha trovato una soluzione:
 - Documentazione della Soluzione:
Viene inserita una descrizione dettagliata della risoluzione, con eventuali note operative o procedure da seguire in caso di problemi simili in futuro.
 - Notifica al Cliente:
Viene inviata una comunicazione al cliente per confermare la risoluzione e, se necessario, per richiedere un feedback sulla soluzione adottata.
- Dopo la verifica della soddisfazione del cliente:
 - Cambio di Stato a "Risolto" e poi "Chiuso":
Il ticket passa prima allo stato "Risolto" e, successivamente, viene chiuso definitivamente dopo un periodo di controllo o a seguito del feedback positivo del cliente.
 - Archiviazione:
Tutte le informazioni e le comunicazioni relative al ticket vengono archiviate per eventuali consultazioni future e per garantire la tracciabilità delle operazioni.
 - Integrare una Knowledge Base che facilita il supporto sia per gli agenti che per gli utenti finali, favorendo il self-service e riducendo i tempi di risposta.

6.2. INTRODUZIONE ALLA CLASSIFICAZIONE DEI TICKET

La classificazione dei ticket in OTRS/ZNUNY è un aspetto fondamentale per garantire una gestione organizzata e efficace delle richieste. Attraverso la classificazione, è possibile:

- Instradare correttamente i ticket ai team o alle code più adatte.
- Definire le priorità e i tempi di risposta in base all'urgenza e all'impatto.
- Monitorare e analizzare le performance del servizio, grazie alla possibilità di generare report e statistiche basate su categorie, priorità e stati.
- Automatizzare processi di routing, escalation e aggiornamento grazie a regole e workflow definiti in base alla classificazione.

6.2.1. Elementi Chiave della Classificazione

La classificazione dei ticket in OTRS/ZNUNY si basa principalmente su tre elementi: **1.**

Servizi

- Definizione:
I servizi sono utilizzate per identificare l'argomento o il tipo di richiesta. Ad esempio, le categorie possono includere "Assistenza Tecnica", "Supporto Amministrativo", "Richieste Commerciali", ecc.
- Modalità di Configurazione:
 - Creazione Gerarchica: È possibile definire servizi e sottoservizi per ottenere una classificazione più granulare.
 - Utilizzo nel Routing: I servizi possono essere usati per instradare automaticamente i ticket a code specifiche o a determinati gruppi di operatori.

- Applicazione nei Report: Permettono di generare statistiche dettagliate per analizzare quali aree generano più richieste e dove focalizzare gli interventi migliorativi.

2. Priorità

- **Definizione:**
La priorità indica il livello di urgenza e l'impatto della richiesta. I livelli tipici possono essere "1 Molto Bassa", "2 Bassa", "3 Normale", "4 Alta" o "5 Molto Alta".
- **Modalità di Configurazione:**
 - **Impostazione Manuale:** Gli operatori possono selezionare la priorità al momento della creazione del ticket, in base all'urgenza percepita.
 - **Assegnazione Automatica:** Attraverso regole di business e workflow, il sistema può assegnare una priorità automatica in base a parole chiave, categoria, o altri criteri (ad esempio, un ticket proveniente da un cliente VIP può essere automaticamente classificato come "Alta").
 - **Integrazione con SLA:** Le priorità sono strettamente collegate agli accordi di livello di servizio (SLA), definendo i tempi di risposta e risoluzione da rispettare. **3. Stati**
- **Definizione:**
Gli stati indicano la fase del ciclo di vita del ticket. I principali stati comprendono:
 - **Nuovo:** Il ticket è stato appena creato e non ancora preso in carico.
 - **Aperto:** Il ticket è attualmente gestito da un operatore.
 - **In Attesa di promemoria:** Il ticket è in attesa di ulteriori informazioni (da parte del cliente o di altri reparti)
 - **Chiuso:** Il ticket è stato chiuso definitivamente dopo la conferma della risoluzione.
- **Modalità di Gestione:**
 - **Aggiornamento Manuale:** Gli operatori aggiornano lo stato del ticket in base alle fasi della gestione.
 - **Automazione:** Tramite workflow, è possibile automatizzare il passaggio da uno stato all'altro in risposta a determinati trigger (ad esempio, un ticket non aggiornato per un certo periodo può passare automaticamente allo stato "In Attesa" o essere segnalato per escalation).

6.2.2. Modalità Operative per la Classificazione dei Ticket

Classificazione Manuale

- **Inserimento Dati Durante la Creazione:**
Quando un ticket viene creato (sia tramite portale self-service, email o manualmente dagli operatori), l'utente che lo crea (o l'operatore) seleziona la categoria e la priorità appropriate dal menu a tendina o da altre interfacce predisposte.
- **Aggiornamento da Parte degli Operatori:**
Durante la gestione del ticket, gli operatori possono aggiornare o modificare la classificazione se, ad esempio, si rende necessaria una revisione in base a ulteriori dettagli emersi.

Classificazione Automatica

- **Regole di Routing e Filtri:**
OTRS/ZNUNY consente di definire regole basate su criteri specifici:
 - **Parole Chiave nel Testo:** Se nel testo del ticket sono presenti termini specifici, il sistema può assegnarlo automaticamente a un determinato servizio o coda.
 - **Email e Mittente:** I ticket provenienti da determinati indirizzi email possono essere classificati automaticamente con una priorità maggiore o indirizzati a code specifiche.
- **Workflow Automatizzati:**

Utilizzando i workflow, è possibile definire condizioni che, al verificarsi di determinati eventi (ad esempio, un ticket con priorità "5 Molto Alta" rimasto inattivo per un certo tempo), attivano azioni come l'aggiornamento dello stato o l'invio di notifiche di escalation.

Integrazione con Altri Sistemi

- Sincronizzazione con Sistemi Esterni:
OTRS/ZNUNY può essere integrato con altri sistemi (CRM, LDAP, ERP, ecc.) per importare informazioni che possono influire sulla classificazione. Ad esempio, dati anagrafici o lo storico del cliente possono determinare la priorità di un ticket.
- Utilizzo di API:
Le API REST o SOAP di OTRS/ZNUNY permettono a sistemi esterni di inviare o aggiornare ticket con le classificazioni già predisposte, garantendo coerenza tra le diverse piattaforme aziendali.

4.1.7 Benefici della Classificazione dei Ticket

- Efficienza e Rapidità:
Una corretta classificazione permette di indirizzare rapidamente le richieste al team giusto, riducendo i tempi di attesa e aumentando la rapidità di intervento.
- Monitoraggio e Reporting:
Analizzando i ticket per categoria, priorità e stato, è possibile ottenere una panoramica chiara del volume di richieste, individuare colli di bottiglia e prendere decisioni informate per migliorare i processi interni.
- Standardizzazione del Servizio:
La classificazione garantisce che ogni ticket segua un percorso standardizzato, facilitando l'adozione di best practice e riducendo il rischio di errori operativi.
- Automazione e Scalabilità: Le regole di classificazione automatica, integrate nei workflow, riducono l'intervento manuale e permettono di gestire un elevato volume di richieste in maniera scalabile.

6.3. INTRODUZIONE ALL'AUTOMAZIONE DEI WORKFLOW IN OTRS/ZNUNY

L'automazione dei workflow in OTRS/ZNUNY è una funzionalità chiave che permette di standardizzare e semplificare il percorso di un ticket lungo tutto il suo ciclo di vita, riducendo l'intervento manuale e garantendo coerenza nelle operazioni. Tramite i workflow automatizzati è possibile:

- Accelerare le risposte: Azioni automatiche in risposta a determinati eventi (ad esempio, notifiche e aggiornamenti di stato).
- Garantire il rispetto degli SLA: Monitorando e agendo in caso di ritardi o mancate risposte.
- Ottimizzare il routing e l'assegnazione: Inoltrando i ticket alla coda o all'operatore più adatto in base a regole predefinite.
- Documentare le operazioni: Registrando ogni passaggio e azione intrapresa, utile per audit e analisi.

6.3.1. Componenti di un Workflow Automatizzato

Un workflow in OTRS/ZNUNY si compone di vari elementi chiave:

Trigger (Attivatori)

I trigger sono gli eventi che innescano l'esecuzione di un workflow. Alcuni esempi includono:

- Creazione di un ticket: L'inserimento di un nuovo ticket nel sistema.
- Cambio di stato: Ad esempio, il passaggio da "Nuovo" a "Aperto".

- Aggiornamento delle informazioni: Modifica della priorità o dell'assegnazione.
- Scadenza di un tempo prestabilito: Ad esempio, un ticket che rimane inattivo per un certo numero di ore.

Condizioni

Le condizioni stabiliscono i criteri che devono essere soddisfatti affinché le azioni automatizzate vengano eseguite. Possono includere:

- Parole chiave nel testo: Se nel titolo o nella descrizione del ticket è presente una determinata parola (es. "critico", "errore di sistema").
- Categoria o priorità: Azioni diverse possono essere applicate a ticket di differenti categorie o priorità.
- Informazioni sul richiedente: Ad esempio, se il ticket proviene da un cliente VIP.

Azioni

Le azioni sono le operazioni eseguite automaticamente quando i trigger si attivano e le condizioni sono soddisfatte. Alcuni esempi di azioni includono:

- Notifiche via e-mail: Inviare un'e-mail agli operatori o ai responsabili.
- Aggiornamento dello stato: Modifica automatica dello stato del ticket (ad es. passaggio a "In Escalation").
- Assegnazione del ticket: Distribuzione automatica del ticket a una coda o a un operatore specifico.
- Aggiunta di note o commenti: Inserimento di commenti standardizzati all'interno del ticket.
- Escalation: Riassegnazione a livelli superiori se il ticket non viene gestito entro un determinato tempo.

6.3.2. Modalità Operative per l'Automazione dei Workflow

Configurazione Tramite Interfaccia Amministrativa

OTRS/ZNUNY mette a disposizione un'interfaccia amministrativa intuitiva, attraverso la quale è possibile:

- Definire e gestire i workflow:
 - Accedere alla sezione "Workflow" o "Automation" (la terminologia può variare in base alla versione) nel pannello di amministrazione.
 - Creare nuove regole specificando trigger, condizioni e azioni da eseguire.
- Modificare regole esistenti:
 - Aggiornare le regole in base a cambiamenti nei processi aziendali o per ottimizzare il flusso di lavoro. • Testare e monitorare:
 - Utilizzare ambienti di test (staging) per verificare il comportamento dei workflow prima della messa in produzione.
 - Monitorare l'esecuzione dei workflow tramite report e log interni per identificare eventuali anomalie.

Personalizzazione Tramite File di Configurazione

Per esigenze avanzate, è possibile intervenire direttamente sui file di configurazione, come ad esempio:

- Kernel/Config.pm:
 - Qui possono essere impostati parametri globali che influenzano il comportamento dei workflow e la gestione dei trigger.
- Script personalizzati:

- Sviluppare script in Perl per estendere o modificare il comportamento standard dei workflow, ad esempio per integrare funzionalità non previste dalla configurazione di base.

Utilizzo di Regole di Business e Filtri

- Definizione di regole dinamiche:
 - Creare filtri basati su parole chiave, categoria o dati specifici del richiedente, che permettano di attivare automaticamente azioni su ticket critici o urgenti.
- Automazione basata su SLA:
 - Integrare il monitoraggio degli SLA nei workflow, in modo che, se un ticket non viene gestito entro il tempo previsto, si attivino azioni di escalation o di notifica.

Integrazione con Sistemi Esterni

- API REST/SOAP:
 - Utilizzare le API di OTRS/ZNUNY per invocare azioni automatizzate da sistemi esterni (CRM, ERP, sistemi di monitoraggio), permettendo una gestione integrata dei processi aziendali.
- Webhooks:
 - Configurare webhooks per ricevere notifiche o aggiornare dati da e verso altri sistemi in tempo reale.

6.3.3. Esempi Pratici di Automazione dei Workflow

Esempio 1: Assegnazione Automatica in Base alla Priorità

- Trigger: Creazione di un ticket.
- Condizione: Il ticket ha la priorità "5 Molto Alta".
- Azione:
 - Assegnare automaticamente il ticket alla coda "Emergenze".
 - Inviare una notifica immediata via e-mail al responsabile del team di supporto.
- Risultato: Il ticket viene gestito con priorità elevata, riducendo i tempi di risposta.

Esempio 2: Escalation Automatica

- Trigger: Un ticket rimane nello stato "Aperto" per più di 24 ore.
- Condizione: Il ticket non ha ricevuto aggiornamenti in questo lasso di tempo.
- Azione:
 - Inviare una notifica ai supervisor e riassegnare il ticket a un operatore senior.
- Risultato: Il problema viene evidenziato e gestito tempestivamente, migliorando il rispetto degli SLA.

Esempio 3: Suggerimenti della Knowledge Base

- Trigger: Durante l'aggiornamento di un ticket, l'operatore inserisce determinati termini chiave.
- Condizione: Il sistema riconosce la presenza di termini legati a problematiche comuni.
- Azione:
 - Visualizzare in tempo reale suggerimenti e articoli dalle FAQ collegati al contenuto del ticket.
 - Permettere all'operatore di allegare direttamente tali articoli al ticket per una soluzione rapida.
- Risultato: Migliora l'efficienza del supporto e favorisce la risoluzione autonoma dei problemi da parte degli operatori.

6.4. INTRODUZIONE ALLE FAQ IN OTRS/ZNUNY

Le FAQ (Frequently Asked Questions) rappresentano una componente strategica della Knowledge Base di OTRS/ZNUNY. Esse raccolgono le domande ricorrenti e le risposte standardizzate relative a problemi comuni, procedure e richieste frequenti. Le FAQ sono fondamentali per:

- Ridurre il carico di lavoro del supporto: Gli utenti possono trovare autonomamente le risposte senza dover aprire un ticket.
- Standardizzare le risposte: Fornendo risposte coerenti e validate, le FAQ assicurano uniformità nel modo in cui le informazioni vengono comunicate.
- Migliorare l'esperienza utente: Sia gli operatori che gli utenti finali hanno accesso a soluzioni rapide, utili e costantemente aggiornate.

6.4.1. Cosa si può fare con le FAQ in OTRS/ZNUNY

Le FAQ in OTRS/ZNUNY offrono una serie di funzionalità e vantaggi:

Creazione di Contenuti Informativi

- Risposte Predefinite:
Le FAQ consentono di creare risposte dettagliate a domande comuni, che possono essere consultate sia dagli operatori che dagli utenti finali.
- Guide e Procedure:
È possibile integrare le FAQ con guide passo-passo e procedure operative per risolvere problemi standard o configurare servizi.
- Materiali Multimediali:
Le FAQ possono includere immagini, screenshot e video che facilitano la comprensione di concetti o operazioni specifiche.

Organizzazione e Indicizzazione

- Categorie e Sottocategorie:
Le FAQ possono essere organizzate in categorie tematiche (es. "Assistenza Tecnica", "Gestione Account", "Problemi di Accesso") per rendere la navigazione più intuitiva.
- Tag e Parole Chiave: L'utilizzo di tag permette di indicizzare e collegare tra loro FAQ correlate, facilitando la ricerca di informazioni specifiche.

Integrazione con il Sistema di Ticketing e il Portale Self-Service

- Suggerimenti Contestuali: Durante la creazione o l'aggiornamento di un ticket, OTRS/ZNUNY può suggerire automaticamente FAQ correlate in base al contenuto o alla categoria della richiesta.
- Accesso al Portale Clienti:
Le FAQ vengono integrate nel portale self-service, consentendo agli utenti di cercare e consultare autonomamente le risposte alle loro domande.
- Aggiornamento Continuo:
Le FAQ possono essere costantemente aggiornate in base al feedback degli utenti e alle nuove problematiche che emergono, garantendo informazioni sempre attuali.

6.4.2. Modalità Operative per Creare e Gestire le FAQ

Creazione degli Articoli FAQ

- Accesso all'Editor FAQ:
Dalla dashboard di OTRS/ZNUNY, accedi alla sezione dedicata alla gestione delle FAQ. Qui è possibile:
 - Creare un Nuovo Articolo FAQ:

Inserisci un titolo descrittivo (ad es. "Come resettare la password") e definisci la domanda in modo chiaro.

- **Struttura del Contenuto:**
Ogni FAQ dovrebbe includere:
 - Domanda: Una formulazione chiara e diretta della problematica o richiesta.
 - Risposta: Una risposta dettagliata che spieghi la soluzione o il procedimento, possibilmente suddiviso in passaggi o punti chiave.
 - Elementi Multimediali (opzionali): Immagini, video o diagrammi che illustrino il procedimento.
- **Utilizzo del Linguaggio e della Formattazione:** È consigliabile utilizzare un linguaggio semplice e diretto. L'editor integrato consente di applicare formattazioni (grassetto, elenchi puntati, tabelle) per rendere il contenuto facilmente leggibile.

Organizzazione e Classificazione delle FAQ

- **Categorie e Sottocategorie:**
Durante la creazione, assegna l'articolo a una o più categorie predefinite, in modo da facilitare l'indicizzazione. Ad esempio, le FAQ relative a "Problemi di Accesso" potrebbero essere collocate in una sottocategoria specifica.
- **Tagging:**
Associa tag rilevanti (es. "password", "login", "autenticazione") che aiuteranno sia gli utenti che il sistema a trovare e suggerire articoli correlati.

Revisione e Approvazione

- **Flusso di Lavoro per la Pubblicazione:**
Implementa un processo di revisione in cui le FAQ vengono controllate da esperti o responsabili prima della pubblicazione. Ciò garantisce che le informazioni siano accurate e aggiornate.
- **Versioning e Cronologia:**
È importante mantenere una cronologia delle modifiche per ogni FAQ, in modo da poter risalire alle versioni precedenti e comprendere l'evoluzione delle informazioni.

Integrazione e Aggiornamento Continuo

- **Suggerimenti Contestuali durante la Gestione dei Ticket:**
Configura il sistema in modo che, in fase di gestione dei ticket, gli operatori ricevano suggerimenti basati sulle FAQ pertinenti, riducendo i tempi di risoluzione.
- **Accesso nel Portale Self-Service:**
Integra la sezione FAQ nel portale clienti, con un motore di ricerca intuitivo che permetta agli utenti di trovare facilmente le risposte alle loro domande.
- **Feedback e Miglioramento Continuo:**
Prevedi meccanismi per raccogliere feedback dagli utenti e dagli operatori, in modo da poter aggiornare e migliorare le FAQ in base alle esigenze emergenti.

6.4.3. Best Practices per una Gestione Efficace delle FAQ

- **Chiarezza e Sintesi:**
Formula le domande in maniera chiara e le risposte in modo sintetico, senza tralasciare i dettagli operativi essenziali.
- **Aggiornamento Periodico:**
Pianifica revisioni regolari delle FAQ per aggiornare le informazioni e rimuovere contenuti obsoleti.
- **Strutturazione Coerente:**

Utilizza una struttura uniforme per tutte le FAQ (titolo, domanda, risposta, elementi multimediali) per facilitare la consultazione.

- Monitoraggio dell'Uso:
Utilizza dashboard e report per monitorare quali FAQ sono più consultate e quali potrebbero necessitare di un aggiornamento o di una maggiore visibilità.
- Coinvolgimento degli Operatori:
Incoraggia il team di supporto a suggerire nuove FAQ o a modificare quelle esistenti in base alle domande più frequenti ricevute durante la gestione dei ticket.

6.5. INTRODUZIONE ALLA REPORTISTICA IN OTRS/ZNUNY

La reportistica in OTRS/ZNUNY rappresenta uno strumento fondamentale per analizzare e monitorare le attività del sistema di ticketing. Essa permette di:

- Misurare le performance del servizio di supporto (tempi di risposta, risoluzione, volume dei ticket, ecc.).
- Identificare tendenze e colli di bottiglia, aiutando a ottimizzare i processi interni.
- Monitorare il rispetto degli SLA (Service Level Agreements) e valutare l'efficacia delle azioni di escalation.
- Fornire informazioni strategiche per decisioni di business e per il miglioramento continuo del servizio.

6.5.1. Cosa Si Può Fare con la Reportistica in OTRS/ZNUNY

OTRS/ZNUNY offre diverse funzionalità e tipologie di report, tra cui: **Report**

Operativi

- Volume dei Ticket: Analisi del numero di ticket aperti, chiusi, in attesa, e in escalation in determinati periodi.
- Tempi di Risoluzione: Report che misurano il tempo medio di risposta e risoluzione per coda, operatore o categoria.
- Andamento degli SLA: Monitoraggio del rispetto degli SLA, evidenziando eventuali violazioni e ritardi.
- Performance delle Code e degli Operatori: Valutazione dell'efficienza dei team e dei singoli operatori, ad esempio, numero di ticket gestiti, tempi medi di intervento e tassi di chiusura.

Report Strategici

- Analisi delle Tendenze: Identificazione dei picchi di richieste e delle problematiche ricorrenti, utile per pianificare risorse e formazione.
- Report per il Management: Dashboard sintetiche con indicatori chiave di performance (KPI), utili per monitorare l'efficacia complessiva del servizio di supporto.
- Report di Customer Satisfaction: Raccoglimento e analisi dei feedback dei clienti per valutare la qualità del servizio offerto. **Report Personalizzati**
- Filtri e Parametri: Possibilità di creare report su misura, definendo filtri per data, coda, categoria, priorità e stato del ticket.
- Query SQL e Script: Per chi ha competenze tecniche, è possibile eseguire query dirette sul database di OTRS/ZNUNY e generare report personalizzati o esportare i dati in formati esterni (CSV, Excel, ecc.).

6.5.2. Modalità Operative per Generare e Gestire i Report in OTRS/ZNUNY

Report Predefiniti e Dashboard Integrate

- Dashboard Interattive: OTRS/ZNUNY integra strumenti per visualizzare dashboard che mostrano grafici e indicatori in tempo reale, facilitando il monitoraggio quotidiano.
- Report Standard: Numerosi report preconfigurati permettono di ottenere subito informazioni su volumi di ticket, SLA, performance delle code e dei singoli operatori.

Utilizzo dell'Interfaccia Amministrativa

- Modulo Reportistica: Accedendo all'area amministrativa, gli amministratori possono selezionare i report disponibili, applicare filtri e impostare i parametri (ad esempio, intervallo di date, categorie specifiche, priorità, ecc.).
- Esportazione Dati: I report possono essere esportati in formati standard (CSV, Excel, PDF) per ulteriori analisi o per la condivisione con il management. **Personalizzazione Avanzata**
- Definizione di Query Personalizzate: Gli amministratori e gli sviluppatori possono scrivere query SQL personalizzate per estrarre dati specifici dal database di OTRS/ZNUNY. Questo permette di creare report ad hoc che rispondono a esigenze particolari.
- Script e Automazioni: Utilizzando script in Perl o integrazioni con tool esterni di BI (Business Intelligence), è possibile automatizzare l'estrazione e l'elaborazione dei dati, integrandoli in sistemi di reportistica aziendale. **Integrazione con Altri Strumenti**
- API e Web Services: OTRS/ZNUNY mette a disposizione API REST/SOAP che consentono di estrarre dati e integrarli in dashboard esterne o sistemi di monitoraggio, favorendo un approccio centralizzato alla reportistica.
- Strumenti di Analisi Esterni: I dati estratti da OTRS/ZNUNY possono essere importati in strumenti di BI (ad esempio, Tableau, Power BI) per analisi più sofisticate e visualizzazioni personalizzate.

6.5.3. Best Practices per la Reportistica in OTRS/ZNUNY

- Definizione degli Obiettivi: Stabilire quali KPI e metriche sono più rilevanti per il team e per il management, per concentrare l'analisi sulle informazioni critiche.
- Monitoraggio Continuo: Configurare dashboard in tempo reale per mantenere sempre sotto controllo i flussi di lavoro e intervenire tempestivamente in caso di anomalie.
- Revisione Periodica: Pianificare revisioni regolari dei report per verificare la coerenza dei dati e aggiornare i parametri in base alle evoluzioni dei processi interni.
- Formazione degli Utenti: Assicurarsi che gli operatori e i responsabili abbiano una formazione adeguata sull'utilizzo degli strumenti di reportistica, in modo da poter interpretare correttamente i dati e prendere decisioni informate.
- Documentazione e Versioning: Mantenere una documentazione aggiornata delle configurazioni e delle query personalizzate, in modo da garantire la continuità operativa e facilitare eventuali aggiornamenti futuri.

6.6. REQUISITI HARDWARE

I requisiti hardware di OTRS/ZNUNY possono variare in funzione del carico di lavoro previsto (numero di ticket, volume di traffico, numero di operatori simultanei, ecc.), ma esistono alcune linee guida generali.

6.6.1. Processore (CPU)

- Ambienti di Test e Piccole Installazioni:
 - Processore multi-core (almeno 2 core) può essere sufficiente.

- Ambienti di Produzione o Ad Alto Carico:
 - È consigliabile utilizzare processori multi-core più performanti (4 core o più) per gestire in modo efficiente il carico elevato e garantire tempi di risposta rapidi.

6.6.2. Memoria (RAM)

- Installazioni di Piccole Dimensioni:
 - Minimo 2 GB di RAM, tenendo conto che l'uso di OTRS/ZNUNY in ambienti di test o in piccoli reparti non richiede molte risorse.
- Ambienti di Produzione:
 - È raccomandato un minimo di 4 GB di RAM, ma per ambienti con un alto volume di ticket o con integrazioni complesse (ad es. workflow avanzati, reportistica intensiva) si consiglia di utilizzare 8 GB o più.

6.6.3. Spazio su Disco

- Storage:
 - Lo spazio su disco deve essere dimensionato in base al numero di ticket, allegati e log generati.
 - Requisiti Minimi: Alcuni GB per l'installazione base e i file di configurazione.
 - Ambienti di Produzione: È preferibile utilizzare sistemi di storage scalabili (SSD o RAID) e prevedere politiche di backup regolari per gestire l'archiviazione dei dati e garantire la continuità operativa.

6.6.4. Connettività e Rete

- Larghezza di Banda:
 - Una connessione di rete stabile e con sufficiente banda è essenziale, soprattutto se OTRS/ZNUNY viene utilizzato in un ambiente con numerosi operatori o in integrazioni con sistemi esterni.
- Rete Interna:
 - È importante che il server su cui gira OTRS/ZNUNY sia configurato per una comunicazione efficiente con il database, il server web e altri sistemi integrati.

6.7. REQUISITI SOFTWARE

- OTRS/ZNUNY è progettato per essere flessibile e personalizzabile e può essere installato su diverse piattaforme. I requisiti software coprono il sistema operativo, il server web, il linguaggio di programmazione, il database e altre componenti necessarie.

6.7.1. Sistema Operativo

- Linux/UNIX:
 - OTRS/ZNUNY è ampiamente utilizzato su distribuzioni Linux (come Debian, Ubuntu, Red Hat, CentOS, SUSE) e sistemi UNIX, che garantiscono una maggiore stabilità e sicurezza per ambienti di produzione.

6.7.2. Server Web

- Apache:
 - Apache HTTP Server è comunemente utilizzato, spesso in combinazione con mod_perl, che consente un'integrazione efficiente con il codice Perl di OTRS/ZNUNY.
- Nginx:

- In alternativa, Nginx può essere configurato per funzionare come server web, anche se potrebbe richiedere configurazioni aggiuntive per l'esecuzione degli script CGI/Perl.

6.7.3. Linguaggio di Programmazione

- Perl:
 - OTRS/ZNUNY è principalmente sviluppato in Perl. È necessario installare una versione aggiornata di Perl (tipicamente Perl 5.8 o superiore) e assicurarsi che siano presenti tutti i moduli necessari, che possono essere installati tramite CPAN o il package manager della distribuzione.

6.7.4. Database

OTRS/ZNUNY supporta diversi sistemi di gestione di database (DBMS):

- MySQL/MariaDB:
 - Tra i DBMS più utilizzati per l'installazione di OTRS/ZNUNY, grazie alla loro diffusione e alla facilità di gestione.
- PostgreSQL:
 - Un'altra opzione robusta, particolarmente adatta a ambienti che richiedono maggiore conformità agli standard SQL.
- Oracle:
 - È possibile utilizzare anche Oracle, soprattutto in contesti aziendali di grandi dimensioni che richiedono un'integrazione con sistemi già esistenti.

I parametri di connessione (DSN, nome utente, password, charset) devono essere configurati correttamente nel file di configurazione principale (tipicamente Kernel/Config.pm).

6.7.5. Altre Componenti e Librerie

- Moduli Perl:
 - Numerosi moduli Perl sono necessari per garantire il funzionamento di OTRS/ZNUNY (ad es. DBI, DBD::mysql o DBD::Pg, MIME::Tools, ecc.). - È importante verificare la compatibilità e installare le versioni richieste.
- SSL/TLS:
 - Se si desidera garantire comunicazioni sicure (sia per l'accesso al portale che per le connessioni API), è necessario configurare il server web con certificati SSL/TLS.
- Altre Dipendenze:
 - Alcuni componenti di OTRS/ZNUNY possono richiedere librerie o strumenti specifici per l'invio di email (SMTP), per il logging e per la gestione dei file di configurazione.

6.8. INSTALLAZIONE

Per installare Znuny è necessario:

- Un database a tua scelta:
 - Mysql 8.0 o mysql 8.4
 - Mariadb 10.3 o più recenti
 - Postgresql 12 o più recenti
 - Oracle 19 o nuovo
- Un server web (Apache)
- Alcuni moduli perl aggiuntivi, a seconda della distribuzione che si sta utilizzando
- Selinux disabilitato quando si utilizza Red Hat Linux, Rocky Linux, ecc.

Le istruzioni di installazione menzionano le distribuzioni Linux. Menzionando Debian ci si riferisce a Debian, Ubuntu e Simile. Menzionando Rhel ci si riferisce a Red Hat Enterprise Linux, Rocky Linux, Alma Linux e Simile. Le distro più utilizzate principalmente sono Debian 12 e Rocky Linux 9.

Per le distribuzioni basate su RHEL, Selinux deve essere disabilitato o impostato su permissive. Alcuni pacchetti di base sono necessari prima di procedere all'installazione vera e propria. Ciò include il server Web, il database (MariaDB in questo caso), cpanminus per installare ulteriori moduli perl e TAR per estrarre i sorgenti.

Comandi per RHEL: `dnf update -y dnf install -y epel-release httpd mariadb mariadb-server cpanminus gcc make dnf-plugins-core tar bash-completion perl-core crb enable`

Comandi per Debian:

`apt update -y apt install -y apache2 mariadb-client mariadb-server tar bash-completion`

Installazione RPM per sistemi basati su RHEL:

`dnf install -y https://download.znuny.org/releases/RPMS/rhel/7/znuny-7.x.x-xx.noarch.rpm`

L'installazione dai sorgenti per sistemi basati su Debian necessita di diversi passaggi:

Scarica Znuny `cd /opt wget`

`https://download.znuny.org/releases/znuny-latest-7.1.tar.gz`

Estrai tar `xfz znuny-latest-7.1.tar.gz`

Crea un symlink `ln -s /opt/znuny-x.x.x/opt/znuny`

Aggiungi utente per Debian/Ubuntu
`useradd -d /opt/znuny -c 'Znuny user' -g www-data -s /bin/bash -M -N znuny`

Copia Config.pm `cp /opt/znuny/Kernel/Config.pm.dist /opt/znuny/Kernel/Config.pm`

Imposta i permessi
`/opt/znuny/bin/znuny.SetPermissions.pl`

```
# Come utente Znuny – crea i cronjobs iniziali
su - znuny cd var/cron
for foo in *.dist; do cp $foo `basename $foo .dist`; done
```

Installazione dei moduli Perl

In base alla distribuzione, ci sono diversi modi per installare i moduli necessari.

Alcuni dei moduli Perl necessari sono installati durante l'installazione dell'RPM su sistemi RHEL, occorre solo aggiungere quelli mancanti.

```
yum install -y "perl(Moo)" "perl(Text::CSV_XS)" "perl(YAML::XS)" "perl(ModPerl::Util)"
"perl(Mail::IMAPClient)" "perl(JSON::XS)" "perl(Encode::HanExtra)"
"perl(Crypt::Eksblowfish::Bcrypt)" "perl(Data::UUID)" "perl(Date::Format)"
"perl(Date::Time::TimeZone)" "perl(Date::Time)" "perl(DBD::mysql)" "perl(DBI)" "perl(Hash::Merge)"
"perl(Net::LDAP)" "perl(Net::DNS)" "perl(Template)" "perl(Template::Stash::XS)"
"perl(XML::LibXML)" "perl(XML::LibXSLT)" "perl(XML::Parser)"
"perl(Spreadsheet::XLSX)" "perl(Package::Stash)"
```

```
cpam JavaScript::Minifier::XS CSS::Minifier::XS iCal::Parser
```

Sistemi Debian:

```
apt -y install libapache2-mod-perl2 libdbd-mysql-perl libtimedate-perl libnet-dns-perl libnet-ldap-perl
libio-socket-ssl-perl libpdf-api2-perl libsoap-lite-perl libtext-csv-xs-perl libjson-xs-perl libapache-
dbiperl libxml-libxml-perl libxml-libxslt-perl libyaml-perl libarchive-zip-perl libcrypt-eksblowfish-perl
libencode-hanextra-perl libmail-imapclient-perl libtemplate-perl libdatetime-perl libmoo-perl
bashcompletion libyaml-libyaml-perl libjavascript-minifier-xs-perl libcss-minifier-xs-perl libauthen-
saslperl libauthen-ntlm-perl libhash-merge-perl libical-parser-perl libspreadsheet-xlsx-perl libdata-
uuidperl
```

6.9. CONFIGURAZIONE DEL DATABASE

MySQL e MariaDB necessitano di alcune modifiche di configurazione. Se si utilizza PostgreSQL, si può saltare questo passaggio

Crea un nuovo file di configurazione per MariaDB:

Sistemi RHEL: vi /etc/my.cnf.d/znuny_config.cnf

Sistemi Debian: vi /etc/mysql/mariadb.conf.d/50-znuny_config.cnf

Contenuto per il file di configurazione:

```
[mysql] max_allowed_packet=256M
```

```
[mysqldump] max_allowed_packet=256M
```

```
[mysqld]
```

```
innodb_log_file_size = 256M
```

```
max_allowed_packet=256M
```

Riavvia il database MariaDB per applicare le modifiche:

```
systemctl restart mariadb
```

6.10. CONFIGURAZIONE DEL SERVER WEB

Sistemi RHEL: La configurazione di Apache è già attiva se si utilizza l'installazione RPM; per l'installazione da sorgenti, crea un collegamento simbolico:

```
ln -s /opt/znuny/scripts/apache2-httpd.include.conf /etc/httpd/conf.d/znuny.conf
```

Abilita il modulo MPM prefork:

```
sed -i '/^LoadModule mpm_event_module modules/mod_mpm_event.so/s/^#/'  
/etc/httpd/conf.modules.d/00-mpm.conf sed -i '/^#LoadModule mpm_prefork_module  
modules/mod_mpm_prefork.so/s/^#/' /etc/httpd/conf.modules.d/00-mpm.conf
```

Sistemi Debian: Per abilitare la configurazione di Apache di Znuny, è necessario creare un collegamento simbolico al file di configurazione incluso:

```
ln -s /opt/znuny/scripts/apache2-httpd.include.conf /etc/apache2/conf-available/znuny.conf
```

Abilitare i moduli e la configurazione necessari di Apache:

```
a2dismod mpm_event a2enmod  
mpm_prefork headers filter perl a2enconf  
znuny
```

Avvia / Riavvia il server web per applicare le modifiche:

```
systemctl restart httpd | systemctl restart apache2
```

Ora avvia l'installer web per la prima configurazione:

<http://HOSTNAME/znuny/installer.pl>

In questa fase viene anche configurato il primo collegamento con mariadb e la creazione del database.

6.11. CONFIGURAZIONE DI AVVIO

Occorre abilitare il server web e il database per avviarsi in automatico ad ogni avvio del server.

Sistemi RHEL: `systemctl enable mariadb httpd`

Sistemi Debian: `systemctl enable mariadb apache2`

Attivazione Cron Znuny

Passa all'utente znuny e avvia il crontab.

6.12. CREAZIONE DEGLI UTENTI

Durante la prima fase di configurazione viene creato l'utente root@localhost con il quale viene effettuato il primo vero e proprio accesso, successivamente si potrà creare l'utente o gli utenti designati a svolgere il ruolo di amministratori di sistema, cioè coloro che poi creeranno gli altri utenti e gestiranno la creazione e l'automazione del workflow.

Questo è il primo passaggio dove il team dell'helpdesk dovrà individuare le figure professionali che ricopriranno questo incarico.

Di seguito la spiegazione passo-passo di come creare un utente:

- **Accedi come Amministratore**
 - Login: Effettua l'accesso al sistema utilizzando un account con privilegi amministrativi.
 - Accesso all'area amministrativa: Una volta loggato, clicca sul menu "Admin" per accedere al pannello di gestione.
- **Naviga nella Sezione Utenti**
 - Gestione Utenti: Nel pannello di amministrazione, cerca e seleziona la voce "Agenti".
- **Avvia la Creazione di un Nuovo Utente**
 - Nuovo Utente: Clicca sul pulsante "Aggiungi Agente".
- **Compila il Modulo di Creazione.** Verrai presentato con un modulo in cui dovrai inserire i dettagli dell'utente. Ecco i campi principali da compilare:
 - Username (Login): Inserisci il nome utente che l'utente utilizzerà per accedere al sistema.
 - Password: Imposta una password conforme alle regole di sicurezza del sistema (alcune installazioni possono richiedere criteri minimi di complessità).
 - Nome e Cognome: Inserisci il nome reale dell'utente.
 - Indirizzo Email: Fornisci un indirizzo email valido; questo può essere utile per notifiche e per il recupero della password.
 - Lingua Preferita: Se applicabile, seleziona la lingua in cui l'utente vedrà l'interfaccia.
 - Altre Informazioni: A seconda della configurazione del tuo OTRS/ZNUNY, potresti avere campi aggiuntivi come il Dipartimento, la Firma, le Preferenze personali o altre impostazioni specifiche.
- **Assegna Gruppi e Ruoli**
 - Gruppi/Code: È importante assegnare l'utente al gruppo o ai gruppi corretti, poiché i gruppi determinano i permessi e l'accesso alle code/ticket.
 - Ruoli: Se il sistema prevede una gestione basata su ruoli, seleziona il ruolo appropriato (ad esempio "Supervisore" ecc.).
- **Salva le Impostazioni**
 - Conferma Creazione: Dopo aver compilato tutti i campi necessari e aver verificato le impostazioni, clicca su "Salva" per finalizzare la creazione dell'utente.
- **Verifica e Configurazioni Aggiuntive**
 - Verifica: Una volta creato l'utente, controlla nella lista utenti per confermare che il nuovo utente sia presente e che tutte le impostazioni siano corrette.
 - Autenticazione Esterna: Per quanto riguarda gli utenti clienti viene utilizzato un sistema di autenticazione esterno basato su LDAP. Per cui gli utenti vengono creati e gestiti tramite il sistema esterno e OTRS/ZNUNY si limita a interrogare il database LDAP per aggiungere gli utenti clienti.

6.13. CONFIGURAZIONE DEI WORKFLOW

6.13.1. Code

Le code sono il punto di ingresso per i ticket e permettono di organizzare le richieste in base a criteri quali reparto, tipologia o priorità.

Creazione di una Nuova Coda

- Accesso all'Area Amministrativa:
Accedi a OTRS/ZNUNY con un account amministrativo e vai nella sezione accessibile tramite il menu "Admin".
- Naviga alla Gestione delle Code:
Cerca la voce "Code" (o "Queues") nel pannello di amministrazione.
- Creazione della Coda:
 - Clicca su "Aggiungi coda" o "New Queue".
 - Compila i campi richiesti, come:
 - Nome della Coda: Un identificativo univoco (es. "Assistenza Tecnica", "Helpdesk Commerciale").
 - Descrizione: Una breve spiegazione dello scopo della coda.
 - Gruppo Associato: Associa la coda a un gruppo specifico di operatori.
 - Priorità/Ordine: Imposta l'ordine o la priorità se necessario.
- Configurazioni Avanzate:
 - Orari di Lavoro: Imposta eventuali orari di servizio che potrebbero influire sul calcolo degli SLA.
 - Regole di Accesso: Definisci chi può visualizzare, modificare o chiudere i ticket presenti in quella coda.
 - Routing e Assegnazioni Automatiche: Se disponibile, configura le regole che smistano automaticamente i ticket verso operatori specifici in base a criteri predefiniti.

Gestione delle Code

- Modifica: Puoi modificare i dettagli di una coda in qualsiasi momento dall'area amministrativa.
- Monitoraggio: Utilizza i report e i dashboard di OTRS/ZNUNY per verificare il flusso dei ticket per ogni coda e intervenire in caso di congestioni o problemi.
- Archiviazione/Eliminazione: Se una coda non è più necessaria, valuta se disattivarla in modo da mantenere pulita l'architettura del sistema.

6.13.2. Servizi

I servizi consentono di raggruppare le richieste che appartengono a specifici ambiti di assistenza (ad esempio, "Supporto Hardware", "Assistenza Software") e di applicare regole dedicate (SLA, notifiche, ecc.).

Creazione di un Nuovo Servizio

- Accesso all'Area Amministrativa:
Dal pannello admin, seleziona la voce "Servizi" (o "Services").
- Nuovo Servizio:
 - Clicca su "Inserisci un Servizio".
- Inserisci i dettagli richiesti:
 - Nome e Descrizione: Identifica chiaramente il servizio.
 - Associazione a Code/Operatori: Specifica se il servizio deve essere collegato a determinate code o gruppi di operatori.
- Parametri Specifici:

- **Priorità e Impatto:** Imposta i parametri che influenzeranno la gestione del ticket.
- **Regole di Escalation e Notifica:** Configura eventuali regole personalizzate che si applicano solo ai ticket appartenenti a quel servizio.

Gestione dei Servizi

- **Modifica:** Puoi aggiornare le informazioni e le regole del servizio in base all'evoluzione delle esigenze aziendali.
- **Reportistica:** Verifica le performance e il rispetto degli SLA associati ai vari servizi tramite i report interni.

6.13.3. SLA (Service Level Agreements)

Gli SLA definiscono i tempi e le condizioni di risposta e risoluzione dei ticket, contribuendo a garantire standard di qualità e tempi di intervento misurabili.

Creazione di un SLA

- **Accesso alla Sezione SLA:**
Dal menu di amministrazione, seleziona "SLA" o "Service Level Agreements".
- **Nuovo SLA:**
 - Clicca su "Aggiungi SLA".
- **Compila i campi necessari:**
 - **Nome:** Un identificativo univoco per lo SLA (es. "SLA Assistenza Tecnica").
 - **Descrizione:** Una breve descrizione delle condizioni previste.
 - **Parametri Temporali:** Definisci i tempi per la prima risposta, la risoluzione e/o eventuali aggiornamenti successivi.
 - **Condizioni di Applicazione:** Specifica a quali code, servizi o tipi di ticket si applica lo SLA.
- **Azioni in Caso di Superamento:**
 - Configura eventuali regole di escalation, che ad esempio possano notificare i responsabili o cambiare la priorità del ticket se i tempi previsti vengono superati.

Gestione e Monitoraggio degli SLA

- **Associazione a Code/Servizi:** Associa lo SLA creato alle code o ai servizi pertinenti, in modo che venga applicato automaticamente ai ticket in ingresso.
- **Report e Dashboard:** Utilizza gli strumenti di monitoraggio di OTRS/ZNUNY per verificare il rispetto degli SLA e identificare eventuali criticità.
- **Revisione Periodica:** In base all'andamento del servizio, rivedi e aggiorna gli SLA per adeguarli alle nuove esigenze o alle best practice del settore.

6.13.4. Notifiche

Le notifiche in OTRS/ZNUNY permettono di automatizzare l'invio di comunicazioni (es. email) in risposta a determinati eventi o cambi di stato dei ticket. **Creazione di una Notifica**

- **Accesso all'Area Notifiche:**
Dal pannello di amministrazione, seleziona "Notifiche" o "Notifications".
- **Nuova Notifica:**
 - Clicca su "Aggiungi notifica" o "New Notification".
- **Configura i seguenti elementi:**
 - **Evento Scatenante (Trigger):** Seleziona l'evento che attiverà la notifica (es. "Nuovo ticket", "Cambio di stato", "Escalation").

- Condizioni: Imposta eventuali filtri (ad esempio, solo per una determinata coda, servizio o priorità).
- Destinatari: Specifica a chi deve essere inviata la notifica (es. l'utente che ha aperto il ticket, il gruppo di supporto, o un indirizzo email esterno).
- Template del Messaggio: Utilizza un template predefinito o creane uno nuovo, includendo variabili dinamiche (come il numero del ticket, il nome dell'utente, la data, ecc.) per rendere il messaggio più informativo.
- Azioni Aggiuntive:
 - Puoi configurare azioni supplementari, come la registrazione nel log, la modifica dello stato del ticket o l'assegnazione automatica a un operatore specifico.

Gestione delle Notifiche

- Test: Prima di attivare una notifica in ambiente di produzione, esegui dei test per verificare che i trigger e i template funzionino come previsto.
- Attivazione/Disattivazione: Le notifiche possono essere attivate o disattivate in base alle necessità. In caso di modifiche ai processi aziendali, aggiorna i template e le condizioni.
- Monitoraggio: Controlla i log e i report per assicurarti che le notifiche vengano inviate correttamente e per identificare eventuali anomalie o necessità di ottimizzazione.

6.14. RIFLESSIONI FINALI E PROSPETTIVE FUTURE

In questa sezione abbiamo esplorato in dettaglio le funzionalità, le configurazioni e le personalizzazioni offerte da OTRS/Znuny, evidenziando come questa piattaforma possa essere adattata alle esigenze specifiche di ogni organizzazione. Dalla gestione delle code e dei servizi, alla definizione degli SLA e alla configurazione delle notifiche, ogni sezione ha illustrato le best practice per sfruttare appieno le potenzialità del sistema, garantendo un supporto tecnico efficiente, trasparente e orientato al continuo miglioramento.

Riepilogo dei Concetti Chiave

- Flessibilità e Personalizzazione:
La capacità di configurare code, servizi, SLA e notifiche permette di modellare il sistema secondo i processi aziendali specifici, garantendo un'implementazione su misura e facilmente adattabile all'evoluzione delle esigenze operative.
- Automazione e Ottimizzazione dei Processi:
Grazie ai workflow automatizzati e alle regole di escalation, OTRS/Znuny riduce il carico di lavoro manuale, assicurando un monitoraggio costante dei ticket e il rispetto degli standard di servizio, con un impatto diretto sull'efficienza operativa.
- Integrazione e Scalabilità:
La piattaforma si integra facilmente con altri sistemi aziendali e supporta un'elevata scalabilità, rendendola adatta sia a realtà di piccole dimensioni che a organizzazioni di grandi dimensioni, con la capacità di gestire grandi volumi di richieste.
- Sicurezza e Gestione degli Accessi:
Un controllo granulare dei ruoli e dei permessi, insieme alla registrazione dettagliata delle operazioni, garantisce la sicurezza dei dati e una tracciabilità completa delle attività, aspetti fondamentali in ambienti che richiedono elevati standard di sicurezza.

7. BUSINESS INTELLIGENCE (APACHE SUPERSET)

7.1. INTRODUZIONE

Apache Superset è una moderna piattaforma di Business Intelligence (BI) open-source che consente agli utenti di esplorare e visualizzare dati interattivamente. Sviluppata originariamente da Airbnb, Superset si è evoluta in una soluzione robusta e altamente scalabile, supportata da una comunità attiva e in costante crescita. Grazie alla sua interfaccia utente intuitiva e alla sua ampia gamma di visualizzazioni, Superset aiuta le aziende a trasformare grandi volumi di dati in informazioni strategiche e fruibili.

Uno dei principali punti di forza di Superset è la sua capacità di integrarsi con numerosi database e datawarehouse, offrendo un ambiente flessibile per la creazione di dashboard personalizzate e analisi avanzate. Questo lo rende una scelta eccellente per aziende e organizzazioni che desiderano adottare un approccio basato sui dati nel proprio processo decisionale.

7.2. FUNZIONALITÀ PRINCIPALI

Apache Superset offre un'ampia gamma di funzionalità che lo rendono uno strumento potente e versatile per la Business Intelligence:

- **Connessione a diverse fonti di dati.** Superset supporta una vasta gamma di fonti di dati, tra cui:
 - Database relazionali come **MySQL, PostgreSQL, Oracle**.
 - Data warehouse moderni come Amazon Redshift, Google BigQuery, Snowflake.
 - File e sorgenti esterne come **CSV, Excel, Google Sheets**.
 - Sistemi basati su **Apache Druid, Presto, Trino e Elasticsearch**. Questa compatibilità consente alle aziende di connettere Superset ai loro sistemi esistenti senza dover migrare i dati.
- **Creazione di dashboard personalizzate:**
 - Gli utenti possono creare dashboard interattive e personalizzate, componendo più visualizzazioni in un'unica interfaccia. Grazie alla funzione di drag-and-drop, è possibile disporre i widget con facilità, migliorando la leggibilità e la fruibilità delle informazioni.
- **Visualizzazioni interattive.** Superset offre una vasta gamma di visualizzazioni, quali:
 - **Grafici a barre, a linee, a torta** per rappresentare dati aggregati.
 - **Mappe geografiche** per analisi spaziali.
 - **Grafici a dispersione, istogrammi e heatmap** per esplorazioni più dettagliate.
 - **Grafici avanzati con D3.js** per visualizzazioni personalizzate. Gli utenti possono applicare filtri interattivi per esplorare i dati in tempo reale.
- **Query SQL avanzate:**
 - Per utenti più esperti, Superset fornisce un'interfaccia SQL che consente di eseguire query personalizzate, aggregazioni complesse e trasformazioni dei dati. Include anche un editor SQL con evidenziazione della sintassi e suggerimenti automatici.
- **Collaborazione e condivisione**

- Superset consente agli utenti di condividere dashboard e report con colleghi e stakeholder. Le dashboard possono essere pubblicate e aggiornate automaticamente, garantendo che tutti abbiano accesso ai dati più recenti.
- **Sicurezza e gestione degli accessi**
 - La piattaforma offre un sistema di autenticazione e autorizzazione basato su ruoli (RBAC), permettendo di definire chi può accedere e modificare i dati. È possibile integrare Superset con sistemi di autenticazione come LDAP, OAuth o SAML.
- **Estensibilità tramite plugin**
 - Superset è altamente estensibile e supporta l'aggiunta di nuove visualizzazioni e connettori dati tramite plugin. Le aziende possono sviluppare funzionalità personalizzate per soddisfare specifiche esigenze analitiche.

7.3. INFRASTRUTTURA NECESSARIA

Per un'implementazione efficace di Apache Superset, è importante garantire un'infrastruttura adeguata. Ecco gli elementi essenziali:

- È necessario un server con risorse sufficienti per gestire il carico di lavoro, con una CPU multi-core, almeno 4GB di RAM e spazio su disco sufficiente per la memorizzazione delle dashboard e dei dati temporanei.
- Linux è il sistema operativo consigliato per l'installazione di Superset, con distribuzioni come Ubuntu, CentOS e Debian.
- Superset richiede un database per memorizzare i metadati, le configurazioni e le connessioni ai dati. PostgreSQL e MySQL sono le opzioni più comuni.
- Un web server come Apache, Nginx o Gunicorn viene utilizzato per servire l'interfaccia web di Superset e gestire le richieste degli utenti.
- Superset è sviluppato in Python e richiede un ambiente Python con dipendenze come Flask, SQLAlchemy e Pandas.

7.4. INSTALLAZIONE E CONFIGURAZIONE

L'installazione e la configurazione di Superset possono essere eseguite in diversi modi.

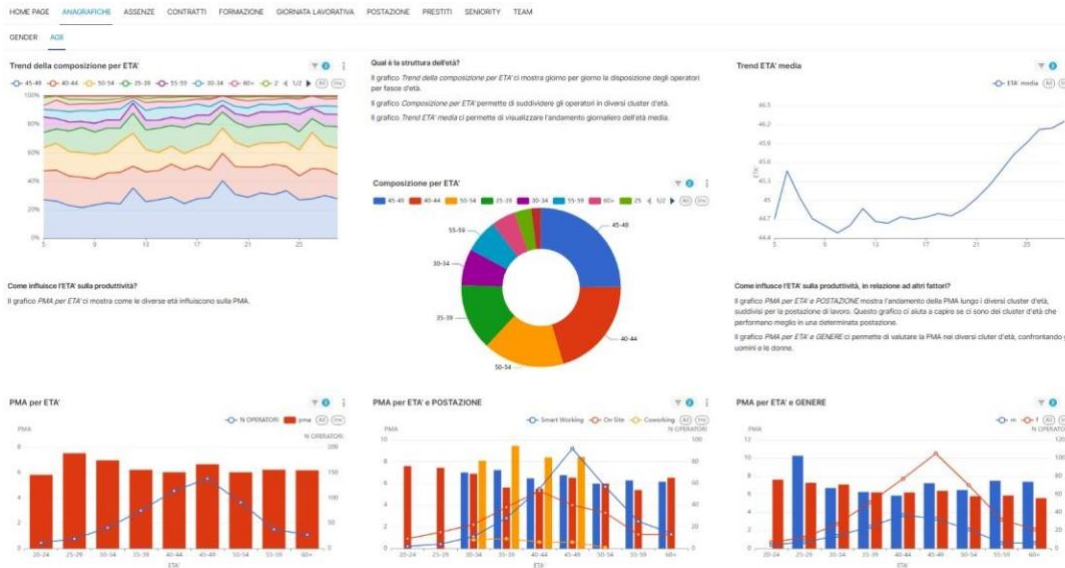
Superset può essere installato utilizzando:

- **Docker** (metodo consigliato per una rapida configurazione).
- **Installazione manuale** tramite pip e ambiente virtuale.
- **Strumenti di provisioning** come Ansible o Terraform.

Dopo l'installazione, è necessario configurare il database di metadati, che conterrà le informazioni relative a utenti, dashboard e connessioni ai dati.

Le fonti dati devono essere aggiunte a Superset specificando connessioni ai database o ai data warehouse aziendali.

Una volta configurate le fonti dati, è possibile creare dataset personalizzati e dashboard interattive tramite l'interfaccia di Superset.



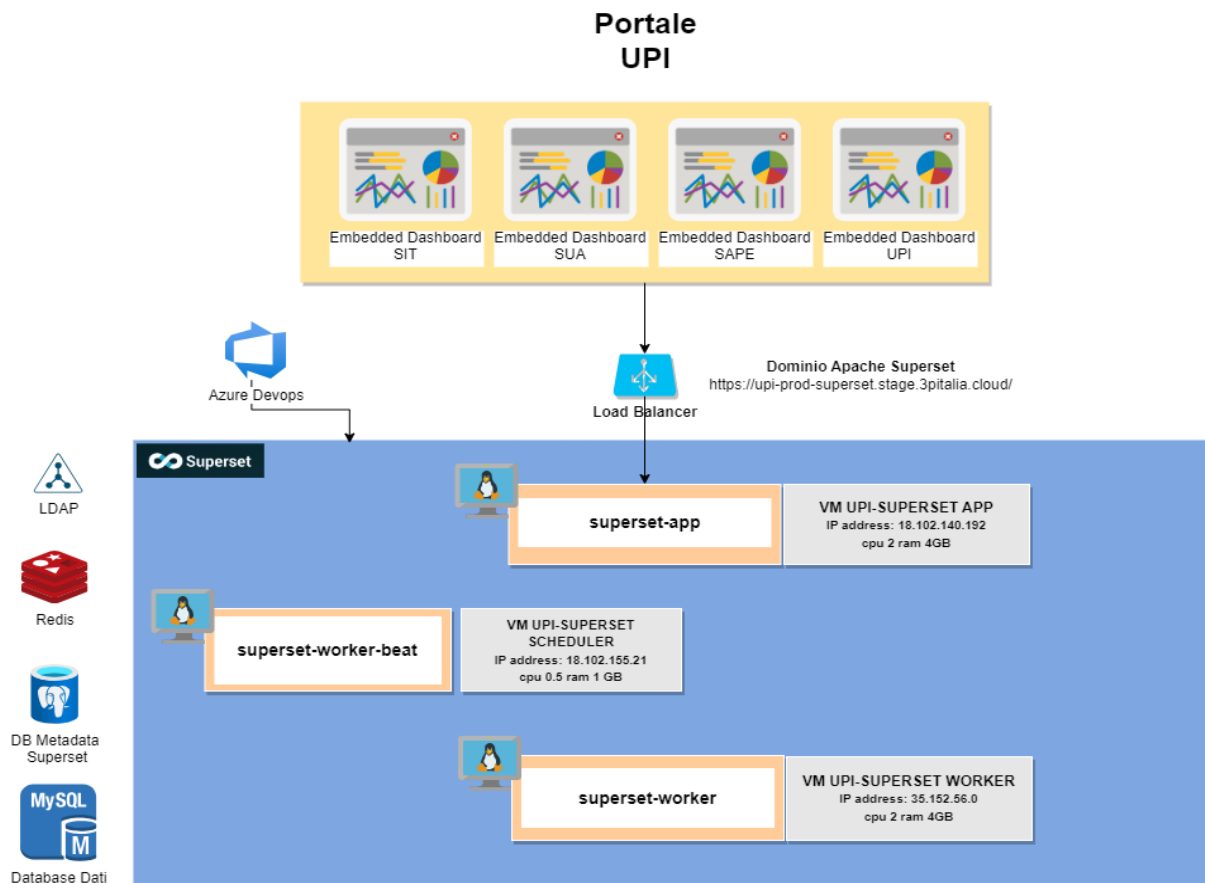
7.5. VANTAGGI DI UTILIZZARE SUPERSET

L'adozione di Apache Superset offre numerosi vantaggi rispetto ad altre soluzioni BI:

- Essendo un software open-source, Superset non ha costi di licenza e beneficia di una grande comunità di sviluppatori che contribuiscono costantemente alla sua evoluzione.
- Può essere personalizzato per adattarsi a diverse esigenze aziendali, consentendo un alto grado di personalizzazione.
- Può gestire grandi volumi di dati ed è ottimizzato per funzionare con data warehouse di livello enterprise.
- Una vasta community supporta il progetto, offrendo documentazione aggiornata, forum di supporto e miglioramenti continui.
- Superset si integra facilmente con strumenti di orchestrazione come **Apache Airflow**, migliorando la gestione dei workflow di analisi dati. Apache Superset è una delle soluzioni di Business Intelligence open source più avanzate e versatili disponibili oggi. Con la sua potenza, flessibilità e facilità d'uso, rappresenta una scelta eccellente per qualsiasi azienda che voglia sfruttare al meglio i propri dati.

7.6. APPROFONDIMENTI RELATIVI ALLA SOLUZIONE PER UPI

7.6.1. Disegno architetturale



7.6.2. Specifiche e lista server

- **UPI-SupersetAPP:**
 - OS: Ubuntu
 - Cpu: 2
 - Ram: 4 GB
- **UPI-SupersetWorker:**
 - OS: Ubuntu
 - Cpu: 2
 - Ram: 4 GB
- **UPI-SupersetScheduler:**
 - OS: Ubuntu
 - Cpu: 2
 - Ram: 4 GB

7.6.3. Setup Server

Collegarsi ad ogni macchina virtuale via SSH ed eseguire i successivi step comuni, ovvero:

- Installazione Docker Engine
- Gestione docker come utente non root
- Installazione Docker compose plugin
- Cloning Azure Devops Repository

Creazione utenza Superset

- Creare utenza applicativa sudo per eseguire tutti i successivi step.

- Dopo aver effettuato la creazione passare alla nuova utenza:

sudo su – superset

Installazione Docker Engine tramite apt repository:

```
sudo apt-get update sudo apt-get install -y ca-  
certificates curl gnupg
```

```
sudo install -m 0755 -d /etc/apt/keyrings curl -fsSL
```

```
https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg sudo chmod a+r  
/etc/apt/keyrings/docker.gpg
```

```
echo \
```

```
"deb [arch="$(dpkg --print-architecture)" signed-  
by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \  
"$(. /etc/os-release && echo "$VERSION_CODENAME)" stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

```
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin  
dockercompose-plugin
```

Controllare se l'installazione ha avuto successo lanciando il seguente comando:

```
sudo docker run hello-world sudo  
docker ps -a
```

Se l'output è simile alla seguente riga allora l'installazione del docker engine è completata:

```
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS  
NAMES  
ec47c8e575be hello-world "/hello" 28 seconds ago Exited (0) 28 seconds ago  
relaxed_vaughan
```

Gestione Docker come utente non root:

Creiamo un gruppo chiamato docker per aggiungere gli utenti all'interno:

```
sudo groupadd docker sudo  
usermod -aG docker $USER  
newgrp docker
```

Prova a lanciare un comando senza sudo come prefisso:

docker ps -a

Installazione Docker Compose plugin:

Installiamo il plugin necessario per i successivi step:

**sudo apt-get update sudo apt-get install -y
docker-compose-plugin docker compose
version**

7.6.4. Azure DevOps repository

Chiedere all'amministratore di Azure DevOps Repository (con il codice di Superset) un Project Access Tokens (nome repository: apache-superset-upi) Clonare il repository in \$HOME : **Azure DevOps token**

- **username:** (in possesso amministratore sistema)
- **token:** (in possesso amministratore sistema)
- **validità:** (in possesso amministratore sistema) **cd \$HOME/**

git **clone**
https://{username}:{token}@dev.azure.com/COMDATADIGITAL/Apache%20Superset/_git/apache-superset-upi

7.6.5. Deploy soluzione Apache Superset

La soluzione Apache Superset è composta dai seguenti servizi:

- **superset_app:** Python WSGI HTTP Server for UNIX (Gunicorn), porta 8088. Servizio deployato nel VM UPI-SupersetAPP
- **superset_worker:** Python Celery worker. Servizio deployato nel VM UPI-Superset Worker
- **superset_worker_beat:** Beat scheduler and worker (tutto in uno) per gestione task schedulati di default da Superset o creati manualmente dagli utenti. Servizio deployato nel VM UPI-Superset Scheduler

Tutti i servizi sono collegati al servizio Postgres (DB per i metadati) ed a Redis per gestione dei async task e gestione della cache.

Configurazione file .env (Docker Compose build)

Modificare il file .env.prod che troverete nella repository aggiungendo tutte le info per la connessione a Postgres e Redis. Salvare il file di fianco a docker-compose-prod.yml con il nome .env

Importante: modificare il file chiedendo il supporto del team di sviluppo. Per impostare le connessioni leggere il seguente doc: https://hackmd.io/@albertogaspari3pitalia/HypwPjG_2 .

Nota: Modifiche sono già impostate, basta copiare .env.prod e rinominarlo .env:

```
cd $HOME/apache-superset-upi/docker-compose cp  
.env.prod .env
```

```
Deploy servizio Superset APP su VM Superset APP chmod u+x $HOME/apache-  
superset-upi/docker-compose/**/* .sh cd $HOME/apache-superset-upi/docker-compose  
docker compose -f docker-compose-prod.yml build docker compose -f docker-compose-  
prod.yml up -d superset superset-init --force-recreate
```

Load Balancer Healthcheck

Aggiungere un healthcheck per monitorare lo stato di salute delle istanze superset_app deployate,
URL path: <http://hostname:8088/health> Se il servizio è UP, l'output sarà OK.

Deploy soluzione su VM Superset Worker

```
chmod u+x $HOME/apache-superset-upi/docker-compose/**/* .sh  
cd $HOME/apache-superset-upi/docker-compose docker  
compose -f docker-compose-prod.yml build  
docker compose -f docker-compose-prod.yml up -d superset-worker --force-recreate
```

Deploy soluzione su VM Superset Scheduler

```
chmod u+x $HOME/apache-superset-upi/docker-compose/**/* .sh  
cd $HOME/apache-superset-upi/docker-compose docker  
compose -f docker-compose-prod.yml build  
docker compose -f docker-compose-prod.yml up -d superset-worker-beat --force-recreate
```

Script di deploy con .sh file

Per semplificare il rilascio della soluzione Superset è stato creato il seguente script .sh, è
necessario creare una variabile d'ambiente SUPERSET_DEPLOY sulla macchina virtuale:

- UPI-SupersetAPP: export SUPERSET_DEPLOY=webapp
- UPI-SupersetWorker: export SUPERSET_DEPLOY=worker
- UPI-SupersetScheduler: export SUPERSET_DEPLOY=beat-scheduler

Inoltre, configurare le variabili d'ambiente associate alla repository Azure DevOps:

- export AZDEVOPS_USERNAME={azure devops username token}
- export AZDEVOPS_TOKEN={azure devops token value} **chiedere i valori**

all'amministratore della repository.

```
#!/bin/bash
```

```
#### Docker Engine & Docker Compose Installation #####
```

```
if ! command -v docker &> /dev/null then echo "Docker non è installato.
```

```
Installazione in corso..." sudo apt-get update sudo apt-get install -y ca-  
certificates curl gnupg
```

```
sudo install -m 0755 -d /etc/apt/keyrings curl -fsSL
```

```
https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg sudo chmod a+r  
/etc/apt/keyrings/docker.gpg
```

```
echo \
```

```
"deb [arch="$(dpkg --print-architecture)" signed-  
by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \  
"$(. /etc/os-release && echo "$VERSION_CODENAME)" stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

```
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin  
dockercompose-plugin
```

```
# check docker engine installation
```

```
sudo docker run hello-world sudo  
docker ps -a
```

```
# run docker command as non root user sudo groupadd docker #  
create the docker group sudo usermod -aG docker $USER # add your  
user to the docker group newgrp docker # activate the changes
```

```
# check command without prefix sudo
```

```
docker ps -a
```

```
else echo "Docker è già  
installato." fi
```

```
if ! command -v docker compose &> /dev/null then
```

```
# docker compose installation sudo apt-get  
update sudo apt-get install -y docker-  
compose-plugin docker compose version
```

```
else echo "Docker Compose è già
installato." fi

#### Azure Devops Repository #####
if [ -d "$HOME/apache-superset-upi" ] then
echo "La cartella esiste. Rimozione in corso..."
sudo rm -r $HOME/apache-superset-upi else
echo "La cartella non esiste." fi

# git clone cd
$HOME
git clone https://$AZDEVOPS_USERNAME:$AZDEVOPS_TOKEN@dev.azure.com/COMDATADIGITAL/
Apache%20Superset/_git/apache-superset-upi

# create .env file
cd $HOME/apache-superset-upi/docker-compose cp
.env.prod .env

# add permission to execute all .sh files in folder
chmod u+x $HOME/apache-superset-upi/docker-compose/**/* .sh

#### DOCKER COMPOSE BUILD AND UP #####
# build services cd $HOME/apache-superset-
upi/docker-compose docker compose -f docker-
compose-prod.yml build

# run services (it depends on where we run command, controlled by SUPERSET_DEPLOY env
variable) if [[ "$SUPERSET_DEPLOY" == "webapp" ]]; then
# run in SUPERSET APP VM docker compose -f docker-compose-prod.yml up -d
superset superset-init --force-recreate elif [[ "$SUPERSET_DEPLOY" == "worker" ]]; then
# run in SUPERSET WORKER VM docker compose -f docker-compose-prod.yml up
-d superset-worker --force-recreate elif [[ "$SUPERSET_DEPLOY" == "beat-
scheduler" ]]; then
# run in SUPERSET SCHEDULER VM docker compose -f docker-compose-prod.yml up -
d superset-worker-beat --force-recreate else
# Genera un errore raise error "Il valore della variabile
d'ambiente non è valido"
```


7.6.6. Approccio Virtual Machine Scale Set (in AWS)

Grazie al rilascio dei container via script .sh, è possibile rilasciare le istanze direttamente all'interno di 3 Scale Set:

- PROD-Upi-supersetSchedulerAutoScaling
- PROD-Upi-supersetWorkerAutoScaling
- PROD-Upi-supersetAppAutoScaling

Ognuna delle 3 componenti è basata su un Modello di avvio costruito su server Ubuntu 22.04 LTS, in cui è stata personalizzato il tipo di istanza, è stato personalizzato il valore di SUPERSET_DEPLOY per rilasciare container diversi nei diversi Autoscale Group.

Le policy per numero iniziale di istanze, numero massimo, numero desiderato e soglia per la partenza di nuove istanze sono definite a livello di singolo Autoscale Group.

Per l'aggiornamento delle istanze si può utilizzare aws cli, consultare la documentazione per l'installazione sui propri sistemi.

E' stata creata su AWS una policy dedicata al refresh istanze con il solo permesso di refresh di qualunque Autoscale Group il cui nome inizi con UPI-Prod, e verranno create utenze personali a richiesta per chi avesse bisogno di eseguire questa operazione.

Il comando principale da utilizzare, una volta impostate le credenziali locali, è aws autoscaling startinstance-refresh <OPTIONS>

Ref:

<https://awscli.amazonaws.com/v2/documentation/api/latest/reference/autoscaling/startinstance-refresh.html>

Comandi AWS CLI

- Refresh istanze per rilascio nuovo codice:

```
aws autoscaling start-instance-refresh --region eu-south-1 --auto-scaling-group-name  
PRODUpi-supersetAppAutoScaling --preferences '{"MinHealthyPercentage": 50,  
"InstanceWarmup": 300, "SkipMatching": false, "ScaleInProtectedInstances":  
"Refresh", "StandbyInstances": "Terminate"}
```

```
aws autoscaling start-instance-refresh --region eu-south-1 --auto-scaling-group-name  
PRODUpi-supersetSchedulerAutoScaling --preferences '{"MinHealthyPercentage": 50,  
"InstanceWarmup": 300, "SkipMatching": false, "ScaleInProtectedInstances":  
"Refresh", "StandbyInstances": "Terminate"}
```

```
aws autoscaling start-instance-refresh --region eu-south-1 --auto-scaling-group-name  
PRODUpi-supersetWorkerAutoScaling --preferences '{"MinHealthyPercentage": 50,  
"InstanceWarmup": 300, "SkipMatching": false, "ScaleInProtectedInstances":  
"Refresh", "StandbyInstances": "Terminate"}
```

Prestare particolare attenzione a MinHealthyPercentage e InstanceWarmup da tarare in funzione delle istanze in esecuzione all'interno del gruppo e del tempo previsto di avvio di una istanza.

- Recuperare i log:

```
aws autoscaling describe-auto-scaling-groups --region eu-south-1 --auto-scaling-group-name  
PROD-Upi-supersetAppAutoScaling
```

```
aws autoscaling describe-auto-scaling-groups --region eu-south-1 --auto-scaling-group-name  
PROD-Upi-supersetSchedulerAutoScaling
```

```
aws autoscaling describe-auto-scaling-groups --region eu-south-1 --auto-scaling-group-name  
PROD-Upi-supersetWorkerAutoScaling
```

7.6.7. Dettagli Architettureali:

PROD-Upi-supersetAppAutoScaling

- min scalabilità: 1 •
max scalabilità:
5
- macchina: - OS:
Ubuntu
 - Vcpu: 2
 - Ram: 4 GB

- Scalabilità rispetto
a CPU e RAM,
misurazione degli
ultimi 5 min
**PROD-Upi-
supersetWorker
AutoScaling**

- min scalabilità: 1 •
max scalabilità:
5
- macchina: - OS:
Ubuntu
 - Vcpu: 2
 - Ram: 4 GB

- Scalabilità rispetto
a CPU e RAM,
misurazione degli
ultimi 5 min

PROD-Upi-supersetSchedulerAutoScaling

- min scalabilità: 1 •
max scalabilità:
1
- macchina: - OS:
Ubuntu
 - Vcpu: 1
 - Ram: 1 GB

- No Scalabilità

Nota: per l'installazione dell'antivirus e grafana agent PROD-Upi-supersetSchedulerAutoScaling ha un dimensionamento maggiore rispetto a quello richiesto.

7.6.8. Connessione ai dati UPI

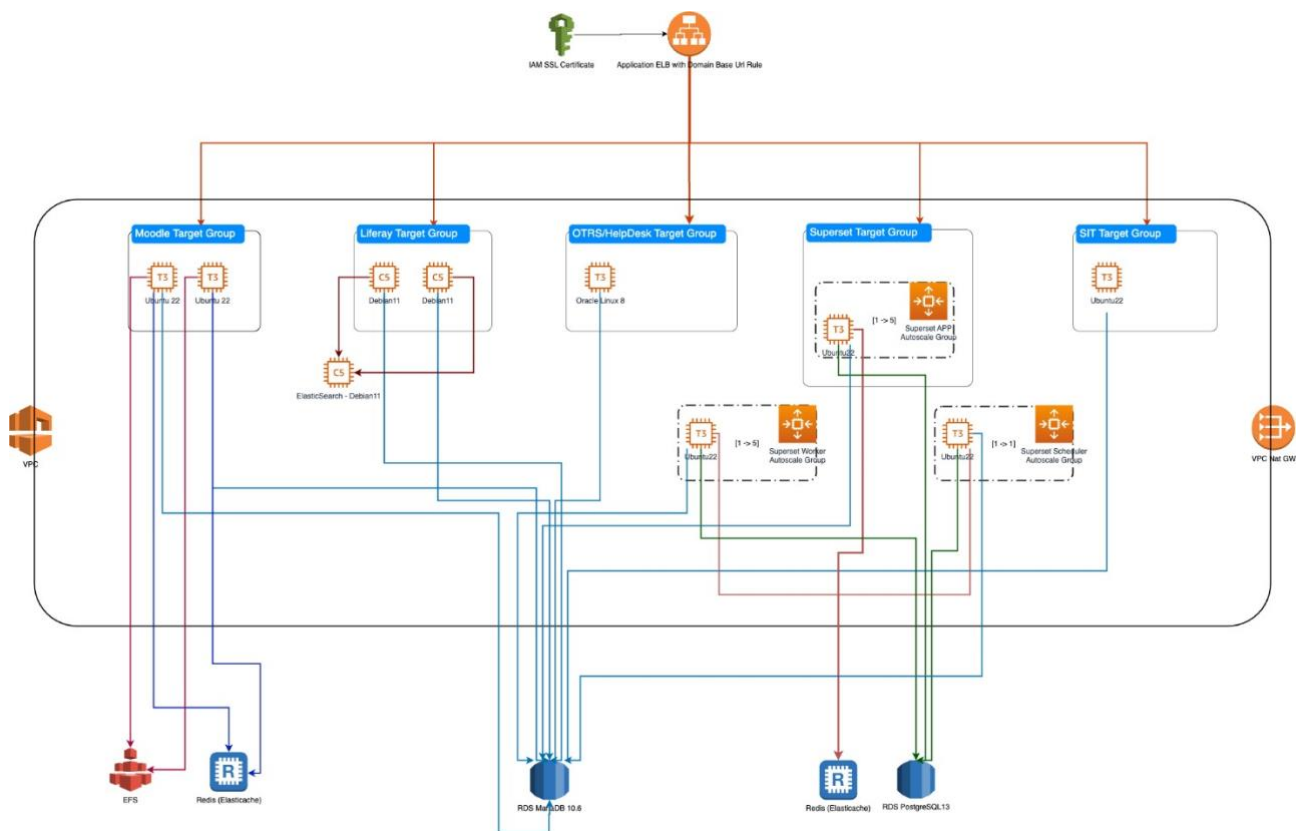
host: [prod-upi-mariadb-1.cye3x1ptfalw.eu-south-](https://prod-upi-mariadb-1.cye3x1ptfalw.eu-south-1.rds.amazonaws.com)

1.rds.amazonaws.com port:3306 db: UPI

Username: (in possesso amministratore sistema) password:

(in possesso amministratore sistema)

8. Infrastruttura AWS



L'architettura della piattaforma Pi.Co. rappresentata nell'immagine mostra un'infrastruttura cloud basata su AWS, progettata per ospitare diverse applicazioni tra cui alcune con bilanciamento del carico, scalabilità automatica e alta disponibilità.

L'ambiente utilizza un Application Load Balancer (ELB) per distribuire il traffico in base alle regole sui domini e gestisce più gruppi di destinazione per differenti applicazioni, ciascuna eseguita su istanze dedicate con specifiche configurazioni. Il backend dell'architettura include database relazionali gestiti (RDS), caching distribuito (Redis), file system condiviso (EFS) e servizi di indicizzazione (ElasticSearch).

8.1. MOODLE TARGET GROUP

Questo gruppo ospita l'applicazione Moodle e prevede la possibilità di eseguire due istanze EC2 basate su Ubuntu 22. Le istanze sono collegate a un file system condiviso EFS per la gestione dei contenuti e utilizzano Redis (Elasticache) per ottimizzare le prestazioni della cache.

8.2. LIFERAY TARGET GROUP

Il gruppo contiene prevede la presenza di due istanze Debian 11 che eseguono Liferay per i moduli orizzontali e i portali. È presente un'istanza aggiuntiva di Elasticsearch per l'indicizzazione e la ricerca avanzata.

8.3. OTRS/HELPDESK TARGET GROUP

Questo gruppo è dedicato alla gestione del sistema OTRS/HelpDesk, eseguito su istanze Oracle Linux 8. Le istanze si connettono a un database RDS MariaDB 10.

8.4. SUPERSET TARGET GROUP

Questo gruppo è distribuito su un cluster autoscalabile che include:

- Un gruppo di istanze EC2 per l'applicazione principale (App Autoscale Group, Ubuntu 22).
- Un gruppo di istanze per i worker di elaborazione dati (Worker Autoscale Group, Ubuntu 22).
- Un'istanza per il servizio di scheduling (Scheduler Autoscale Group, Ubuntu 22). Le istanze sono connesse a un database RDS PostgreSQL 13 e utilizzano Redis (Elasticache) per la gestione della cache.

8.5. SIT TARGET GROUP

Quest'area dell'infrastruttura ospita il verticale SIT eseguito su un'istanza Ubuntu 22, isolata dagli altri gruppi, ma comunque accessibile attraverso il bilanciatore di carico.

L'istanza si connette a un database RDS MariaDB 10.6.

8.6. ALTRI COMPONENTI

- **Application Load Balancer (ELB):** Gestisce il traffico in ingresso e lo indirizza ai diversi target group in base alle regole di dominio.
- **IAM SSL Certificate:** Fornisce la sicurezza crittografica alle connessioni.
- **VPC e NAT Gateway:** Consentono l'isolamento e la comunicazione sicura tra le istanze all'interno della rete cloud.

Segue il dimensionamento dell'infrastruttura AWS dedicata alla Piattaforma Pi.Co. Oltre alla tabella che segue, vanno considerati i servizi accessori a consumo per i servizi messi a disposizione dal cloud provider (es. indirizzi IP statici, load balancer, ...).

Componente	Sizing
PROD-Upi-Moodle1	2VCPU/8GB/40GB
PROD-Upi-Moodle2	2VCPU/8GB/40GB
PROD - Helpdesk UPI	2VCPU/8GB/50GB
PROD - Helpdesk UPI-2	2VCPU/8GB/50GB
PROD-Upi-Liferay1	4VCPU/8GB/100GB
PROD-Upi-Liferay2	4VCPU/8GB/100GB
PROD-Upi-ES	4VCPU/8GB/50GB
PROD-Upi-sit1	2VCPU/4GB/50GB

SERVIZIO DI REALIZZAZIONE DELLA PIATTAFORMA COLLABORATIVA Pi.Co.
 nell'ambito del progetto dell'UPI «Le Province e il sistema dei servizi a supporto dei Comuni»

SupersetAPP	2VCPU/4GB/50GB
SupersetScheduler	2VCPU/2GB/50GB
SupersetWorker	2VCPU/4GB/50GB
prod-upi-mariadb-1	4VCPU/16GB/100GB
prod-upi-postgresql-1-instance-1	2VCPU/16GB/1GB
prod-upi-postgresql-1-instance-1-eu-south-1b	2VCPU/16GB
prod-upi-redis-elasticache (nodo 1)	2VCPU/7GB
prod-upi-redis-elasticache (nodo 2)	2VCPU/7GB
prod-upi-redis-superset (nodo 1)	2VCPU/3GB
prod-upi-redis-superset (nodo 2)	2VCPU/3GB

