# Specifiche di sistema per fornitori di applicazioni

| Nr. versione | Data | Descrizione |
|---|---|---|
| 1.0 | Dic-2013 | Start documento |
| 2.0 | Apr-2016 | Prima revisione |
| 2.1 | Sett-2016 | Aggiornamento revisione |
| 2.2 | Lug-2017 | Aggiornamento revisione |
| 2.3 | Gen-2019 | Aggiornamento revisione |

# 1. Premessa

Lo scopo del documento è quello di definire nel miglior modo possibile il contesto sistemistico nel quale l'applicazione verrà installata. Si tratta essenzialmente di linee guida e buone pratiche, ma anche di introdurre alcuni elementi di valutazione sulla qualità dell'installazione, che hanno una qualche correlazione con i sistemi. Essendo, quello qui descritto, uno "standard" per il Comune, e che per la maggior parte delle applicazioni in uso viene già ampiamente rispettato, chiediamo al Fornitore di porre una particolare attenzione.
In generale, l'applicazione potrà risiedere su uno o più server dedicati, ma anche semplicemente sfruttare una pila software già condivisa con altre applicazioni.
Il presente documento si applica anche nel caso di aggiornamenti tecnologici di applicazioni già esistenti, che abbiano un impatto considerevole sugli aspetti inerenti al Sistema (ad es. l'applicazione necessita di un nuovo web server, application server,..).

# 2. Destinatari

I destinatari del presente documento sono i seguenti:
- il Fornitore;
- il Cliente (Comune di Padova) nelle figure dell'analista/i e del sistemista/i di riferimento.

# 3. Specifiche per l'applicazione

## 3.1 Specifiche Hardware

### 3.1.1 Piattaforma di virtualizzazione

La piattaforma Open Source di riferimento adottata dal Comune di Padova come standard per la virtualizzazione dei server è PVE (Proxmox Virtual Environment) che si basa sulla soluzione KVM (Kernel-based Virtual Machine).
KVM è in grado di far girare istanze virtuali dei seguenti sistemi operativi, senza modificarli: tutte le distribuzioni Linux, BSD server e Sistemi Operativi Windows nelle varie versioni
È una soluzione realizzata dall'Ente e clusterizzata su più nodi.

### 3.1.2 Spazio disco e altre risorse hardware

In un ambiente suddetto è di fondamentale importanza il *provisioning* delle risorse hardware necessarie all'applicazione in oggetto. In questo contesto ci si riferisce in particolare a :
- spazio disco (GB) necessario ai dati;
- spazio disco (GB) necessario al software applicativo;
- n. di virtual CPU richieste;
- memoria RAM (GB) da allocare.

Queste informazioni sono a carico del Fornitore e sono riferite alla sola applicazione da installare.
In assenza di questi dati, verrà fornito uno spazio disco su una partizione/i già esistente/i di dimensione variabile.

## 3.2 Specifiche Software

### 3.2.1 Sistema Operativo

Il sistema operativo, qualora non vi siano particolari esigenze dettate dall'applicazione, è Linux con particolare predilezione per Debian, nell'ultima versione "stable" a 64 bit.

### 3.2.2 Software d'ambiente

Se l'applicazione richiede l'installazione di particolari "pacchetti", il Fornitore deve farne richiesta e l'installazione degli stessi viene concordata col tecnico del Comune.

In linea di principio il comportamento sarà il seguente:
- se si tratta di software della versione già "pacchettizzata" da Debian, verrà privilegiata questa installazione, lasciando tutte le impostazioni di default, compresa la cartella di destinazione;
- se, invece, si rende necessaria l'installazione di pacchetti partendo dal codice sorgente, sarà da preferire la cartella /opt. In questo caso, potrà essere che l'installazione sia a carico del Fornitore, che dovrà possedere tutto il know-how necessario per una corretta installazione e configurazione.

Ogni installazione dovrà essere fatta con opportuni utente/i e gruppo/i "applicativi". Non sono ammesse installazioni di software su cartelle con owner e group che siano "nominativi" (ad es. rossim acronimo di Mario Rossi).

Il fornitore deve verificare, prima della messa in produzione, che l'applicazione esponga all'utenza tutto e solo ciò che è necessario al funzionamento della stessa (solo a fini di esempio: se si utilizza Apache, il fornitore deve interfacciarsi col tecnico del Comune e deve fornire le indicazioni per la configurazione del file sotto /etc/apache2/site-available o /etc/apache2/conf-available per impedire il browsing/listing delle directories sotto /var/www).
Più dettagliatamente, ogni virtual host deve avere un proprio file di configurazione in /etc/apache2/site-available o /etc/apache2/conf-available in modo da isolare le varie configurazioni (un file per ciascun applicativo esposto da apache), queste vanno poi abilitate con i comandi di apache; vanno documentati inoltre eventuali moduli specifici di apache da abilitare normalmente non di default, es. mod_auth_cas.

La impostazione di default su nuove VM Debian che abbiano necessità di Apache, sarà la seguente:
# a2dismod autoindex
# service apache2 restart

### 3.2.3 Demoni e/o servizi

Il Comune di Padova ha adottato Nagios, un'applicazione open source per il monitoraggio di server e risorse di rete. La sua funzione base è quella di controllare nodi, reti e servizi specificati, avvertendo quando questi non garantiscono il loro servizio o quando ritornano attivi.
Il Fornitore deve, quindi, indicare quali sono le porte applicative e i servizi da monitorare. Di tali servizi/demoni devono essere fornite le istruzioni per un arresto e un riavvio, e lo user con cui devono essere eseguite. Inoltre, devono essere avviati automaticamente al boot del sistema, senza che venga richiesta alcuna azione da parte del tecnico del Comune.
Devono, altresì essere indicate, se necessarie, le schedulazioni di attività di tipo *batch* di carattere applicativo, gestite attraverso crontab o Operazioni Pianificate di Windows.
Inoltre, se si tratta di un "cluster applicativo" (più host cooperativi e con funzionalità specifiche), in cui è richiesta una sequenza ben precisa nell'avvio dei vari server componenti il cluster, devono essere realizzati dal fornitore opportuni script che effettuino il boot e/o restart dei sistemi in modo tale da rispettare l'ordine necessario ad un corretto funzionamento. Analoghi script devono prevedere uno spegnimento secondo sequenza, ad esempio per una manutenzione programmata.

### 3.2.4 Backup e Disaster Recovery

Il Comune di Padova ha standardizzato le seguenti procedure in ambito Backup-Restore/Disaster Recovery:
- con periodicità settimanale viene fatto un *dump* dell'intera immagine virtuale del server attraverso gli strumenti nativi messi a disposizione dalla piattaforma di virtualizzazione (Disaster Recovery);
- con periodicità giornaliera viene effettuato un backup delle cartelle applicative e dei dati attraverso il software Open Source Bareos, e la *retention* di tali file o cartelle è garantita per un periodo di due settimane (Backup). Di *default* vi è incluso anche un salvataggio delle cartelle di sistema (/etc, /home, /root).

Il Fornitore deve, perciò, indicare quali sono le cartelle applicative, i dati, ed eventualmente i log da salvare.

### 3.2.5 Sicurezza ed accesso da remoto

La password di root/Administrator del sistema non potrà essere concessa per alcuna ragione al Fornitore, nemmeno durante la fase di avvio dell'applicazione.
In alternativa, è possibile ottenere:
- uno user nominativo (tipicamente formato dal cognome e dalla prima lettera del nome) con password, avente privilegi di amministrazione. In questo modo è possibile per il Comune attivare un Log Management per *loggare* gli accessi, come previsto dalla normativa vigente per gli Amministratori di Sistema.
- Uno user applicativo (es. portale, nettuno,….) con una propria home all'interno della quale viene fatto il *deploy* dell'applicazione da parte del Fornitore.

La scelta dell'uno o dell'altro user è determinata dalla condizione che il server sia rispettivamente dedicato all'applicazione (singolo server/singola application/ singolo fornitore) oppure no.

L'accesso da remoto, ai soli server di pertinenza, è normalmente attivato attraverso la generazione di un certificato OpenVPN nominativo, che sarà inviato al Fornitore, previa compilazione da parte dello stesso del modulo "Richiesta accesso rete dati del Comune di Padova" (v. allegato 2).

### 3.2.6 Visibilità su Internet dell'applicazione

Particolare attenzione agli aspetti di sicurezza deve essere posta dal Fornitore nel caso in cui l'applicazione sia rivolta oltre che agli utenti interni anche, o solamente, a soggetti esterni (cittadini, imprese,…). Il Fornitore dovrà giustificare la piattaforma applicativa prescelta e dovrà indicare le misure messe in atto per scongiurare il più possibile attacchi di tipo informatico al sistema.
Non potranno essere accettate architetture obsolete od obsolescenti particolarmente vulnerabili ad attacchi di tipo *DoS*, (malfunzionamento dovuto ad un attacco informatico in cui si esauriscono deliberatamente le risorse del sistema che fornisce un servizio, come ad esempio un sito web, fino a renderlo non più in grado di erogare il servizio) o di *Defacing* (cambiare illecitamente la home page di un sito web o modificarne, sostituendole, una o più pagine interne).

In caso di applicazione Web tramite tecnologia APACHE2 il fornitore dovrà obbligatoriamente utilizzare almeno i seguenti moduli apache:
- modsecurity2 (libapache-mod-security) (esiste anche per IIS e nginx) con regole di default attivate OWASP ModSecurity Core Rule Set (CRS) (http://spiderlabs.github.io/owasp-modsecurity-crs/)
- modevasive (libapache2-mod-evasive).

Le comunicazioni tra i client e i server su Internet dovranno privilegiare la forma di criptazione dei dati, anche attraverso lo scambio di certificati server SSL e l'accesso protetto da username e password.

L'applicazione che deve essere acceduta anche da Internet potrebbe essere collocata su server in rete interna del Comune, (è ad es. il caso in cui i moduli suddetti non possano essere installati) ma la sua pubblicazione affidata a regole di reverse-proxy tramite tecnologia APACHE2. Il Fornitore deve, in questo caso, garantire piena compatibilità con questa metodologia di pubblicazione e fornire eventuali regole di reverse proxy specifiche per la piattaforma sulla quale il software è stato sviluppato, es. liferay, plone, altro.

### 3.2.7 Svilluppo di software per conto del Comune di Padova

Lo sviluppo di software per conto del Comune di Padova deve rispettare quanto previsto dalle "Linee guida sviluppo software" rilasciate dal Comune stesso.
Tali linee guida sono indirizzate sia ai fornitori del Comune di Padova, che ai propri tecnici interni che partecipano nel processo di sviluppo e gestione del software.
I principali obiettivi di tali linee guida sono:
- fornire uno schema per la corretta pianificazione e gestione delle attività di sviluppo, evoluzione e manutenzione del software
- definire una corretta procedura per il versioning del codice sorgente
- fornire una modalità operativa per implementare la tracciabilità delle attività di sviluppo/bugfix e l'associazione delle attività al versioning del codice
- indicare gli strumenti messi a disposizione dall'ente per l'attività di progettazione e sviluppo
- definire uno standard per la redazione della documentazione afferente il progetto e software/applicazione

Ai sensi del Decreto Legislativo 7 marzo 2005, n. 82 (CAD) con riferimento al Riuso dei programmi informatici, al termine dello sviluppo il codice sorgente e tutta la documentazione (tecnica, di progetto e manualistica) relativa al software dovrà essere consegnato al SS.II.TT (Settore Servizi Informatici e Telematici del Comune).
Il Comune di Padova mette a disposizione gli strumenti che seguono:
- Piattaforma/repository GIT, il codice sorgente dovrà essere "conservato" e gestito mediante il reopistory GIT fornito dal Comune di Padova e ospitato presso i sistemi informatici dell'ente.
Il versioning dovrà rispettare quanto indicato nelle suddette "Linee guida sviluppo software".
Gli ambienti di test/collaudo e produzione dovranno essere integrati con il repository GIT del Comune di Padova al fine di garantirne una corretta gestione del codice sorgente/versioni
- iTracker, portale per la gestione della comunicazione con i fornitori, l'organizzazione e la gestione delle attività (gantt) legate allo sviluppo, evoluzione e messa in produzione di un software. iTracker dovrà inoltre essere utilizzato per la gestione dei ticket inerenti bug/anomalie.
- iWiki, portale per la redazione e gestione della documentazione di progetto, tecnica e manualistica in stile “wikipedia” (seguendo quanto indicato nelle stesse linee guida)
- PARepository/cloud: servizio cloud messo a disposizione del Comune di Padova a supporto dei progetti per l'interscambio di file (ma non gestione codice sorgente o documentazione)

Per ragioni di sicurezza il Comune richiede che i sorgenti, relativi ad applicazioni sviluppate per conto del Comune di Padova, siano conservati nei soli sistemi di proprietà dell'ente (repository GIT e server dell'ente). Al termine dell'attività di sviluppo il Fornitore si impegna a rimuovere materiale ospitato/pubblicato su hosts che non siano in gestione al Comune di Padova.

# 4. Specifiche per il database

## 4.1 Database di riferimento

I Sistemi di Gestione di Base di Dati adottati dal Comune di Padova sono i seguenti:
- Oracle
- PostgreSQL
- Mysql

La preferenza segue l’ordine indicato, poiché il database Oracle è attualmente in una configurazione Real Application Cluster (RAC) e quindi in Alta Disponibilità (HA); mentre così non è per i DBMS PostgreSQL e Mysql.

### 4.1.1 RDBMS Oracle

Versione del database: 11.2.0.4 Standard Edition.

Ad esclusione di alcune particolari applicazioni piuttosto complesse (suite applicative), in cui è stato generato *ad-hoc* un nuovo database (per numerosità di schemi coinvolti, per ragioni prestazionali,…), in tutti i rimanenti casi il Fornitore potrà disporre di un database condiviso con altre applicazioni e su di un server che, di norma, non coincide con il server applicativo.

È questa la ragione per cui il Fornitore deve precisare:
- versione del db;
- dimensionamento di partenza e stima di crescita della/e tablespace (sempre identificative dell'applicazione e mai generiche – es. USERS non è mai utilizzata);
- preferenza sul characterset (altrimenti la scelta è operata dal Comune);
- nome dello schema/i (lunghezza, se possibile entro gli 8 caratteri);
- Grants. Non saranno presi in considerazione per ragioni di sicurezza i seguenti grants: DBA, export full database, import full database, ed in generale tutto ciò che non è strettamente legato allo schema/i in questione;
- eventuali parametri di istanza diversi dal default;
- necessità o meno di un ambiente database di test.

### 4.1.2 PostgreSQL e MySQL

Opzionalmente, il Fornitore che ha sviluppato l'applicazione, può richiedere che il Comune metta a disposizione un database diverso da Oracle. É qui contemplato il caso di una applicazione a bassa criticità, per la quale non siano richiesti particolari accorgimenti atti a garantire una Alta Disponibilità del database, e nemmeno esigenze specifiche di *restore* ad un dato momento temporale già trascorso (Point in Time Recovery).

La scelta sul DBMS di riferimento può, in alcuni casi, essere dettata dal fatto che si vuole realizzare un software utilizzando esclusivamente prodotti "Open Source", anche in un'ottica di "riuso" dell'applicazione (Codice dell'Amministrazione Digitale (CAD) al Capo VI "Sviluppo, acquisizione e riuso di sistemi informatici nelle pubbliche amministrazioni").
A difesa della sicurezza, si chiede che le versioni implementate nelle nuove installazioni siano le più aggiornate possibili.

Rimangono valide le informazioni che il Fornitore deve dare, come indicato al precedente punto 4.1.1, per poter permettere al sistemista del Comune la creazione dello schema.

## 4.2 Sicurezza e Backup

L'accesso diretto al database server dalla sede del fornitore non può essere permessa per ragioni di sicurezza.
Il fornitore dovrà servirsi, eventualmente, dell'application server per creare attraverso una connessione ssh (es. via putty) un tunneling verso il db server, definendo una porta a piacere in localhost. Successivamente, con uno strumento qualunque che produce query (sqldeveloper, squirrel,.....) si creerà una connessione al db attraverso localhost sulla porta prescelta.

Nei database Oracle viene di norma garantito il backup logico e fisico dei dati, attraverso le utility messe a disposizione dalla stessa Oracle. Più precisamente, sono schedulati via crontab o job:
- i dump notturni del db a cadenza giornaliera e con profondità temporale di retention del dato pari ad almeno una settimana (export full database);
- i backup fisici (rman) del database e dei redolog per un eventuale ripristino del tipo *Point in Time*, anch'essi con profondità di almeno una settimana.

Nei database PostgreSQL e MySQL, invece, come suddetto, è garantito solamente il backup logico dei dati, attraverso le utility *pgdump* o *mysqldump*, che possono essere schedulate, a richiesta, anche più volte nel corso della stessa giornata.

Le export dello schema/i proprietario/i sono possibili sui client aziendali del Fornitore; diversamente, le import sui sistemi del Comune vanno sempre concordate col sistemista.

# 5. Requisiti sistemistici dell'Applicazione

## 5.1 Introduzione

Se con l'accezione "requisiti funzionali" si intendono i servizi che il software deve erogare agli utenti, evidenziando le diverse modalità di utilizzo (interazioni) da parte dei possibili attori e gli scenari di utilizzo - elementi essenzialmente di carattere applicativo che esulano dagli scopi del presente documento – non è così per quelli che vengono definiti "requisiti non funzionali", i quali comprendono un'ampia categoria di esigenze che possono essere espresse dagli utenti e dai committenti del software, tra le quali, ad esempio, le "prestazioni" e l'efficienza (vincoli sul tempo di risposta, sull'occupazione di memoria, ecc.), la sicurezza, la usabilità, l'affidabilità, la tecnologia da utilizzare.
Questi "requisiti sistemistici", contrariamente ai primi, sono correlati ai sistemi.

Questo documento si è ispirato allo standard ISO/IEC 9126-1 che descrive un modello per la qualità del software.

## 5.2 Metriche

Vengono qui di seguito indicate alcune metriche che saranno oggetto di valutazione durante la fase di "Collaudo di Sistema" dell'applicazione, che dovrà necessariamente avvenire **prima** della messa in produzione dell'applicativo e dovrà naturalmente avere esito **positivo**. La scheda relativa è all'allegato 2.

### 5.2.1 Efficienza

Questa metrica misura le prestazioni del software in termini di comportamento rispetto al tempo (tempo di risposta di una query, tempo di esecuzione di una transazione, ecc..), di utilizzo delle risorse messe a disposizione dal sistema ( occupazione di memoria, spazio disco,ecc..) e conformità rispetto a standard attesi e/o stabiliti a riguardo.

Per misurare il comportamento dell'applicazione **rispetto al tempo** va rilevato:
- il tempo media necessario al software per eseguire una funzione (es. tempo medio di risposta di una transazione tipo);
- il throughput prodotto nell'unità di tempo (es. n. di record elaborati al secondo);

Per misurare il comportamento dell'applicazione **rispetto all'utilizzo delle risorse** va rilevato:
- l'utilizzo dell'I/O;
- tempo di attesa per l'I/O;
- carico massimo prodotto dall'applicazione sull'I/O;
- utilizzo della memoria nell'esecuzione di uno o più task;
- utilizzo della rete;

### 5.2.2 Manutenibilità

Questa metrica misura il tempo e le risorse necessarie quando sia indispensabile eseguire dei test sull'applicativo.
Dal punto di vista del sistema, si sostanzia nella disponibilità di un ambiente di test (sia applicativo che di database), per poter effettuare test lasciando inalterato l'ambiente di produzione.

### 5.2.3 Adattabilità

Questa metrica misura il comportamento dell'applicazione durante le attività di porting.

- Adattabilità della struttura dei dati (database): è una misura di quanto è facile il porting verso versioni di database superiori, o addirittura verso altri DBMS.

- Adattabilità dell'applicazione all'ambiente hardware: misura quanto l'applicazione è *hardware dependent* e cosa comporta, se possibile, una migrazione su piattaforma differente.
- Adattabilità dell'applicazione all'ambiente software: misura quanto l'applicazione è dipendente da quella precisa versione dei software installati e, se è possibile in futuro, una migrazione su piattaforma software con pacchetti aggiornati.

### 5.2.4 Installabilità

Questa metrica misura l'attività richiesta al tecnico sistemista del Comune per predisporre l'ambiente di installazione.

- Facilità di installazione dei pacchetti software, prerequisito per l'installazione del software applicativo (Ad es. pacchettizzazione std. di Debian con impostazioni di default = molto facile; installazione su user space partendo da codice sorgente, con applicazioni successive di patch e configurazioni specifiche = molto difficile);
- Efficienza di installazione, misura il tempo necessario al sistemista del Comune per effettuare le installazioni di cui sopra;

### 5.2.5 Interoperabilità

È la misura sull'efficacia dello scambio di dati tra l'applicazione ed altri sistemi esterni (ad es. via web services, db link tra database,ecc..).

### 5.2.6 Coesistenza

Questa metrica è molto importante perché è misura di quanto l'applicativo da installare può coesistere con altri prodotti software o con versioni differenti dello stesso componente (Ad es. versioni di JVM differenti).

### 5.2.7 Sicurezza

Queste metriche valutano alcuni aspetti relativi alla sicurezza dal punto di vista informatico.

- Gestione Utenti: è privilegiata quella che fa riferimento al Server LDAP del Comune, in cui sono già presenti tutte le utenze informatiche e vi è già implementata una corretta politica di gestione del cambio password.
- Comunicazione tra client esterni all'Ente e server: deve avvenire attraverso protocolli "sicuri" (es. HTTP attraverso un meccanismo di crittografia di tipo Transport Layer Security (SSL/TLS)).
- Librerie: il Fornitore deve utilizzare le ultime versioni messe a disposizione dal sistema, deve informare il sistemista del Comune se ritiene indispensabile utilizzare versioni più vecchie, in particolare ci si riferisce a importanti *bug* sul sistema coperte da aggiornamenti e/o patch;
- Vulnerabilità: il fornitore dovrebbe indicare all'Ente a che tipo di vulnerabilità è esposta la sua applicazione, in riferimento alla tecnologia utilizzata.
- In caso di web application l'Ente utilizzerà obbligatoriamente tecnologie Open Source di scanning dell'applicazione, come ad esempio:
- skipfish http://code.google.com/p/skipfish/
- w3af http://w3af.org/

  e /o altri scanner per controllare l'esistenza di vulnerabilità che possono comportare compromissioni o leak di informazioni non volute. L'esito di questa scansione è condizione essenziale per la messa in produzione dell'applicazione stessa. Sarà cura del reparto Sistemi del Comune eseguire periodicamente lo scanning e in caso di esito negativo verrà avvisato il fornitore, che dovrà provvedere a mettere in sicurezza il software.

Per esito negativo si intende la presenza di:

- High risk flaws (potentially leading to system compromise),
- Medium risk flaws (potentially leading to data compromise).
- Deve essere rispettato anche quanto descritto ai paragrafi 3.2.2. (ultimo capoverso) e 3.2.7.

### 5.2.8 Gestione dei LOG

Attraverso questa metrica si misura la capacità del Fornitore nella gestione autonoma dei log applicativi, oppure la sua tempestività nel chiedere una gestione congiunta al sistemista del Comune. È il caso ad esempio di file di log che per ragioni dettate da normative vigenti devono essere conservati indefinitamente, o comunque oltre la gestione standard (rotazione e compressione del log precedente). E' in carico al fornitore la gestione della rotazione dei log, che garantisca al sistema la non saturazione delle risorse assegnate.

### 5.2.9 Documentazione a corredo del software

È la metrica attraverso la quale il Comune di Padova misura la qualità della documentazione fornita a corredo del software:
- scheda installazione
- documentazione di progettazione
-documentazione tecnica e sistemistica (descrizione architettura, materiale sistemistico, processo installazione/configurazione, etc)
- manualistica per l'utente "amministratore di sistema"
- manualistica per l'utente finale

Il Fornitore deve produrre e consegnare la documentazione entro i tempi congrui, secondo quanto previsto dal piano di sviluppo per l'applicazione in oggetto.
La consegna della documentazione non dovrà mai avvenire successivamente all'avviamento in produzione dell'applicativo.
Il Comune di Padova fornisce il portale iWiki per la redazione, conservazione e gestione della documentazione. Il portale iWiki e la documentazione dovranno essere redatti secondo quanto indicato nelle "Linee guida sviluppo software".

## *5.3 Allegato 1*

**Padova, _____**

**Spett.le**
**Comune di Padova**
**Settore Servizi Informatici e Telematici**
**Via Frà Paolo Sarpi, 2**
**35138 - Padova**

**Oggetto: Richiesta accesso rete dati del Comune di Padova.**

Il sottoscritto _____
In qualità di _____
della Ditta _____
con sede legale _____

in relazione:
❑ al contratto n. _____ del Settore _____
referente _____ ;
❑ alla collaborazione _____ con il Settore _____
referente _____;
❑ altro _____ con il Settore _____
referente _____;

CHIEDO

la connessione alla rete dati del Comune di Padova per il seguente scopo:

_____
_____
_____
_____

In particolare, **CHIEDO**, per il periodo dal _____ al _____, per n. _____ postazione/i di lavoro, con macaddress _____:

❑ la connessione ad una presa di rete cablata presso la sede comunale di:
Via _____ stanza/uff. _____
❑ accesso ad Internet con IP pubblico;
❑ accesso alla rete aziendale da remoto via Internet;
❑ accesso alla rete aziendale;
❑ accesso alla rete aziendale da IP pubblico _____;
e, conseguentemente, l'abilitazione alle seguenti operazioni:

_____
_____

_____

_____

_____

Comunico che le persone (max 3) che accederanno alla/e postazione/i saranno:

_____

_____

_____

Con la sottoscrizione della presente mi impegno a mettere in atto tutte le misure affinché:

- le credenziali di accesso o specifiche tecniche non siano cedute né divulgate;
- le operazioni effettuate siano strettamente limitate alle finalità sopra descritte;
- sia rispettata tutta la normativa legata alla privacy;
- non vengano effettuati, nel caso di utilizzo Internet, accessi a siti che per contenuti e immagini siano in contrasto con le finalità dichiarate;
- vengano rispettate le norme previste nel Regolamento per l'utilizzo degli strumenti informatici del Comune di Padova, pubblicato in www.padovanet.it

FIRMA

_____

N.B. Il presente modulo deve essere presentato almeno 10 giorni lavorativi prima dell'evento.

**Contatti:**

_____

Tel. _____

Fax _____

E-mail _____

---

**COLLAUDO**

**(Il collaudo dovrà essere effettuato almeno 2 gg lavorativi prima dell'evento descritto nella richiesta)**

Si certifica che tutte le richieste indicate nella domanda sono state controllate e collaudate e ne è stato verificato il buon funzionamento.

Ditta o Referente Informatico

_____

Padova, _____

**N.B. Dopo il collaudo il modulo dovrà essere, restituito debitamente firmato, al Settore Servizi Informatici e Telematici o via fax al numero 049/820.5315 o via e-mail segreteria.sit@comune.padova.it.**

### *5.4 Allegato 2*

**Comune di Padova**

# COMUNE DI PADOVA

**Settore Servizi Informatici e Telematici**
**- Area Sistemi -**

## "Verbale di COLLAUDO Software – Area Sistemi"

**Progetto: " "**
**Prot. n.**

| | |
|---|---|
| CONSEGNA | ☐ |

| | |
|---|---|
| INSTALLAZIONE | ☐ |

## TEST DI SISTEMA[1]

| **METRICA** | VALORE (A=alto, M=medio, B=basso) |
|---|---|
| EFFICIENZA | ☐ |
| MANUTENIBILITÀ | ☐ |
| ADATTABILITÀ | ☐ |
| INSTALLABILITÀ | ☐ |
| INTEROPERABILITÀ | ☐ |
| COESISTENZA | ☐ |
| SICUREZZA | ☐ |
| GESTIONE DEI LOG | ☐ |

| | |
|---|---|
| DOCUMENTAZIONE | ☐ |

Padova,

**ditta:**

---

**COMUNE DI PADOVA**
**Settore SS.II.TT.**

---

[1] I Test Applicativo/funzionali non sono oggetto di tale documento, ma vengono demandati alla relativa Area Applicativa