Introduzione

Traccia dell’esame in breve

Una azienda manifatturiera operante nel settore dell’abbigliamento ha deciso di migliorare la sua rete di vendita utilizzando strumenti tecnologici innovativi. La rete di vendita dell’azienda è costituita da un insieme di agenti/rappresentanti; ad ognuno di essi è assegnato un territorio costituito, in genere, da alcune province limitrofe.   
Le attività svolte dagli agenti sono: la **promozione** dei prodotti dell’azienda madre sul territorio di competenza, la **raccolta** degli ordini sottoscritti dai vari clienti (in genere negozi di abbigliamento), la **trasmissione** degli stessi all’azienda madre. Per svolgere correttamente il loro lavoro, gli agenti hanno bisogno di informazioni generalmente fornite dall’azienda: **disponibilità** dei vari prodotti, **date di consegna** previste per ogni prodotto, **prezzi di vendita dei prodotti**, **eventuali campagne promozionali** in atto su determinati prodotti.   
Poiché tali informazione variano **frequentemente e velocemente**, l’azienda ha pensato di migliorare la comunicazione tra sede centrale ed agenti creando una applicazione informativa **client/server** e fornendo ad ogni agente un portatile. L’applicazione ha lo scopo di fornire agli agenti tutte le informazioni necessarie per svolgere correttamente il proprio lavoro; in particolare, utilizzando la parte client, gli agenti possono conoscere:  
1) La quantità disponibilità di un determinato articolo.  
 2) Il prezzo corrente di vendita di un determinato articolo.  
 3) Lo sconto corrente di vendita di un determinato articolo.  
 4) Un testo che esprime la valutazione commerciale dell’azienda nei confronti di un determinato cliente.  
  
Analisi e raccolta dei requisiti

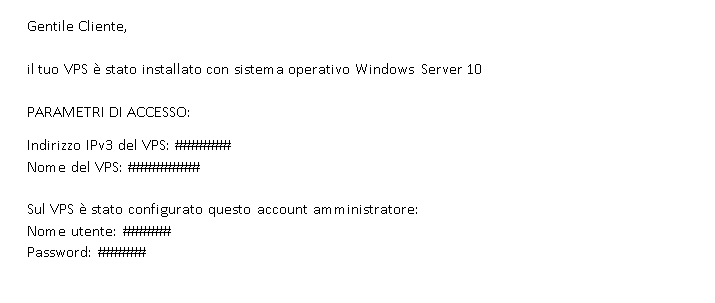
Prima di incominciare ad progettare, abbiamo iniziato ad documentarci su come funziona un business to business e principalmente come vengono gestiti i clienti e amministratori di rete.  
Secondo la nostra documentazione, gli agenti sono professionisti che concludono accordi e contratti (e promuovere) in suo nome ad altri clienti o aziende più piccole e lavorano per territori che prevedono una serie di comuni e città.  
 Il progetto sarà strutturato alloggiato all’interno di una VPS, cioè un computer virtuale, per permettere di tener il server sempre attivo, poiché è essenzialmente un computer che non si spegne mai. Nella VPS inseriremo tutti i nostri strumenti di lavoro (editor, IDE, Web server, database) ed una volta testato e collaudato, andremo ad acquistare un dominio (cioè mascheriamo l’indirizzo IP della macchina in un indirizzo comprensibile all’uomo, come per esempio google.com, facebook.com, wikipedia.org. Inoltre sarà documentato nel file successivo il project charter destinato ai sviluppatori del sistema.  
Le tecnologie utilizzate sono Java e MySQL che verranno spiegati nei successivi paragrafi.  
  
Sistemistica

Infrastruttura sistemistica e di rete.  
Come abbiamo accennato, come prima tappa fondamentale è l’acquisto e configurazione di una VPS, il motivo è molto più semplice del previsto: purtroppo l’hosting, cioè un servizio offerto dalle server farm, sono principalmente in commercio solo ed esclusivamente con PHP e NodeJS (Javascript lato backend, il nuovo colosso del mondo della programmazione e padre di tanti nuovi siti come Netflix e Paypal) e quindi abbiamo preferito (a discapito del prezzo, ma comunque un costo molto basso) comprare questo server virtuale, che oltre al vantaggio di essere accesso perennemente (Eccetto caso di forzatura da parte dell’administrator system o dall’azienda madre), permetterà in qualsiasi momento, in qualsiasi postazione di entrare nel server per assistenza tecnica. Una volta concluso il progetto contenente nella VPS, ogni agente gli verrà assegnato un portatile con una router WI FI, cioè un router portatile di piccole dimensioni con interfaccia USB, per permettere la navigazione sul sito web e poiché ci teniamo alla serietà dei nostri agenti il router WI-FI sarà dotata di firewall, per monitorare le operazioni effettuate e creare delle regole di sistema così da impedire di usufruire strumenti non inerenti all’azienda (come navigare su youtube, twitter, instagram)   
  
Installazioni

Come accedere e configurare ad una VPS

Per semplicità useremo sia come nostro sistema Operativo, che della VPS, Windows Microsoft, così da permettere anche a programmatori non amanti di linea di comando, di essere abbastanza produttivi.  
Ovviamente dobbiamo recarci su siti di vendita di server e useremo OVH, uno dei migliori in Italia. Una volta selezionato la configurazione hardware della nostra VPS (Scelta del processore, modello, storage e ram) e pagato il prodotto per una quantità di tempo, attendiamo la nostra email (Figura 1).

Figura 1



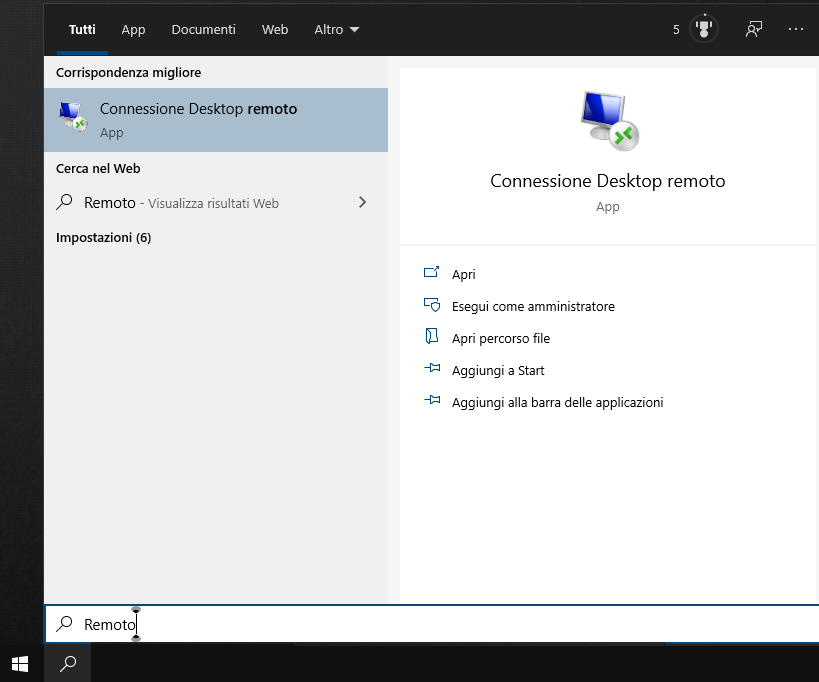
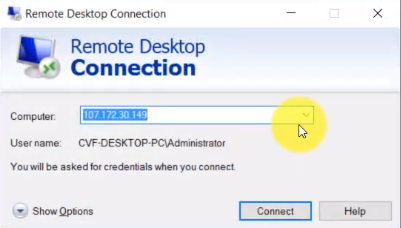
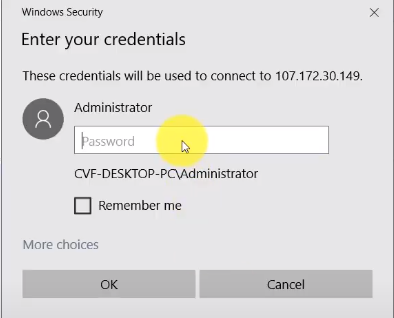
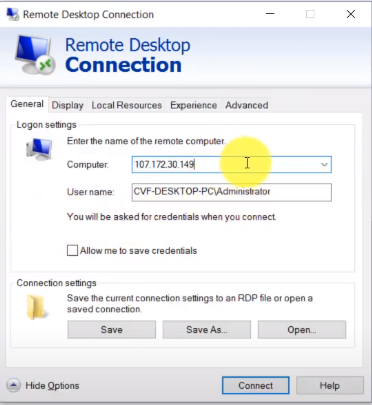
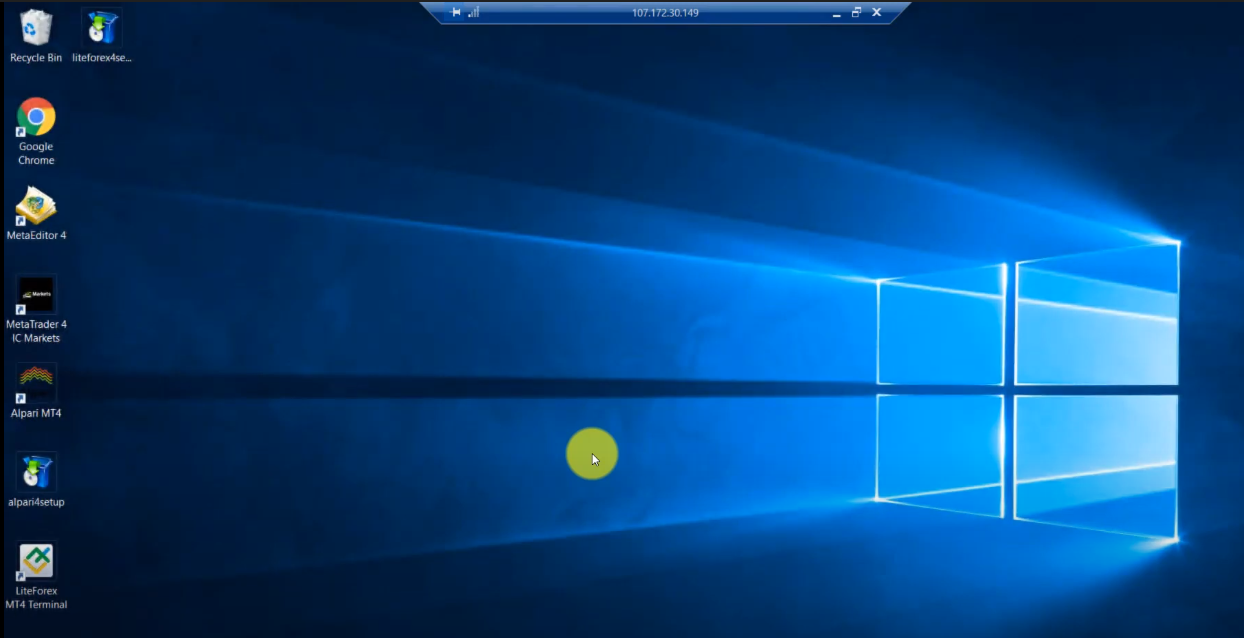
Bene, una volta che siamo in possesso delle nostre credenziali, apriamo la barra della ricerca e connettiamoci da remoto. (Figura 2)

Figura 2

E inserire le credenziali fornite: nella pagina principale dell’applicazione, la textfield da riempire nella label computer, si intende l’indirizzo ip della nostra macchina virtuale ed selezionare il tasto ‘show Options’ o ‘mostra opzioni’. Per ‘user name’ dobbiamo inserire il profilo che l’azienda ci ha forniti. Se non è indicata è di default Administrator. Una volta connesso inserire la password ed finalmente abbiamo fatto accesso al computer virtuale





Installazione di MySQL e XAMPP Tomcat

Il nostro web 2.0 (web dinamico) deve essere alloggiato non solo nella VPS ma precisamente dentro ad web container e quindi abbiamo preferito XAMPP, una piattaforma software multipiattaforma e libera (cioè è possibile essere installato su più sistemi operativi ed open source) contiene all’interno altri strumenti necessari per utilizzare i linguaggi di programmazione PHP, Perl e Java. Con questa installazione ci sarà fornito anche MySQL.

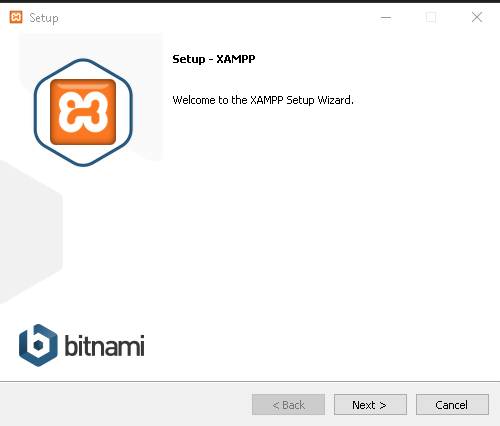
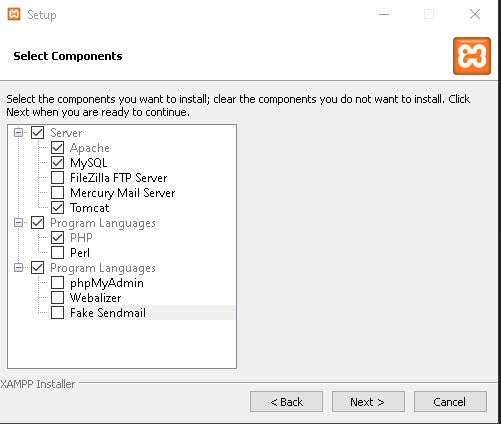
Innanzitutto abbiamo recarci sul sito web ufficiale ([*www.apachefriends.org*](http://www.apachefriends.org)) e scaricare la versione per Windows (Figura 4 A) . Selezionare i checkBox indicati in figura 4 B e selezionare il path (Figura 4 C) della nostra applicazione da installare. Una volta installato vi verrà mostrato il pannello di controllo che ci permette di attivare e disattivare con semplici click il server e/o MySQL. (Figura 4 E)

Figura 4 A

Figura 4 B  


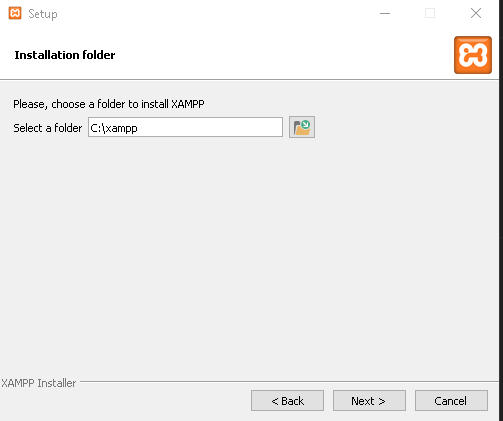
Figura 4 C

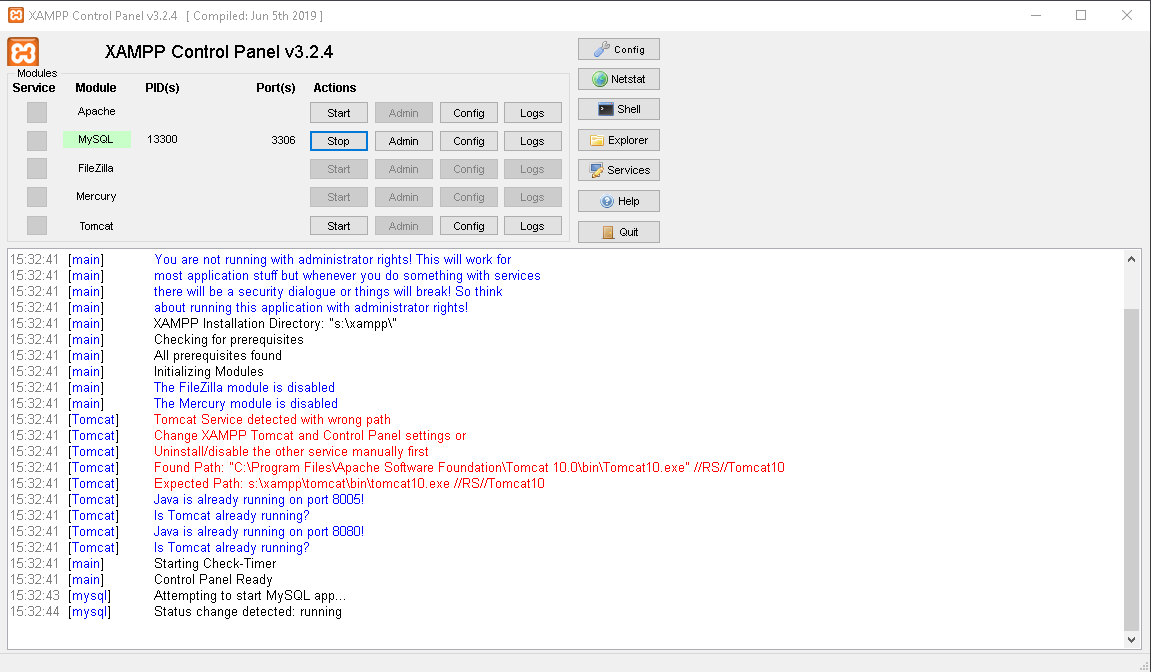
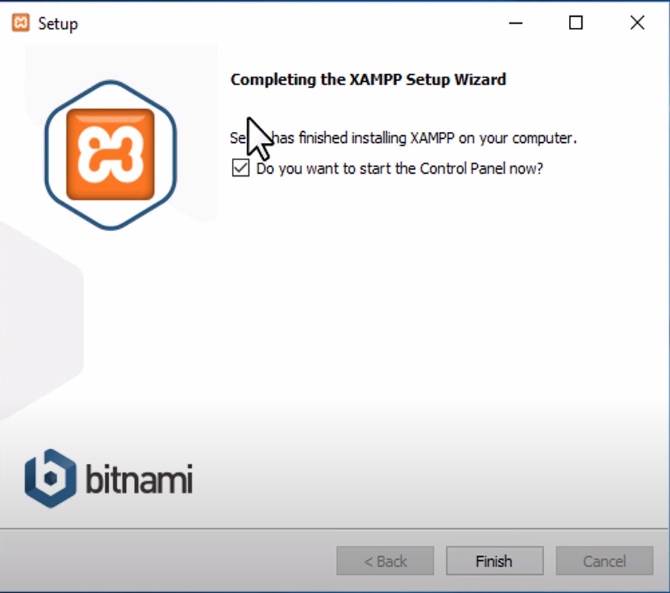
Figura 4 D

Figura 4 E

Configurazione dei NoteBook.

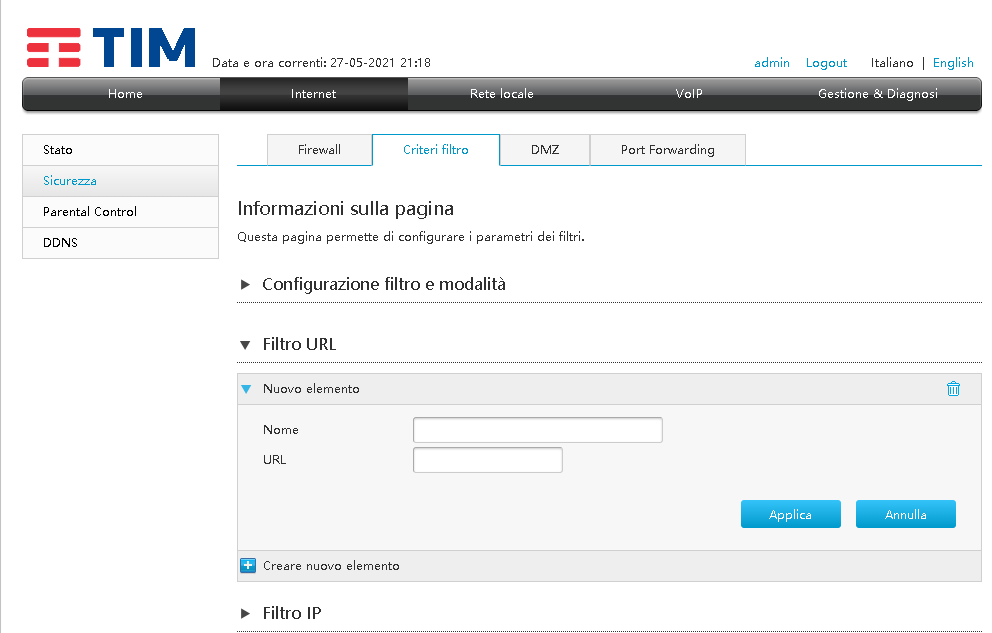
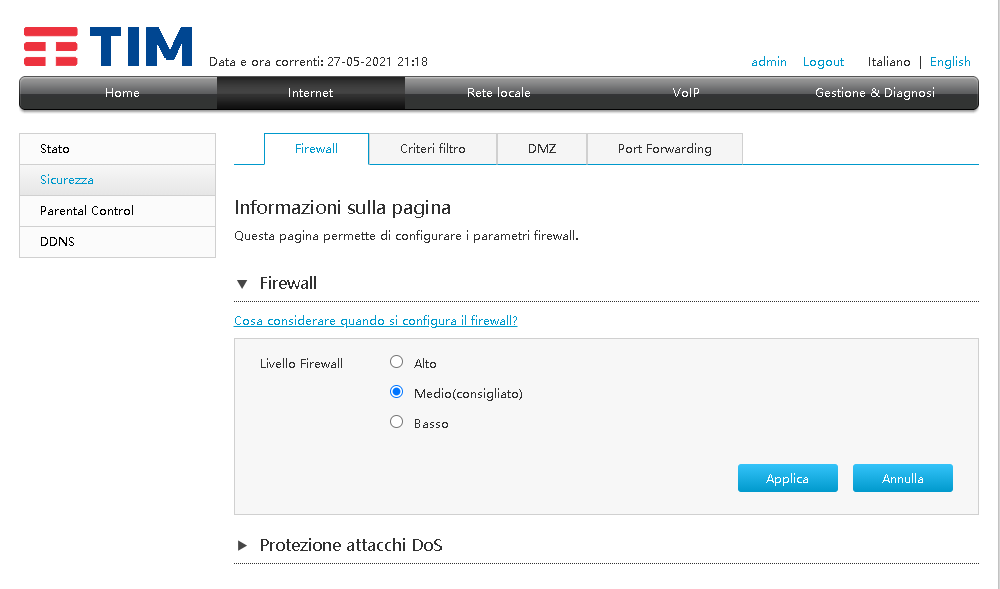
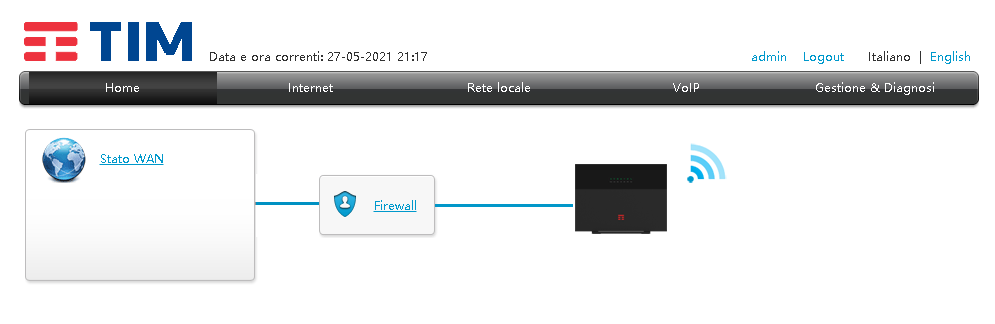
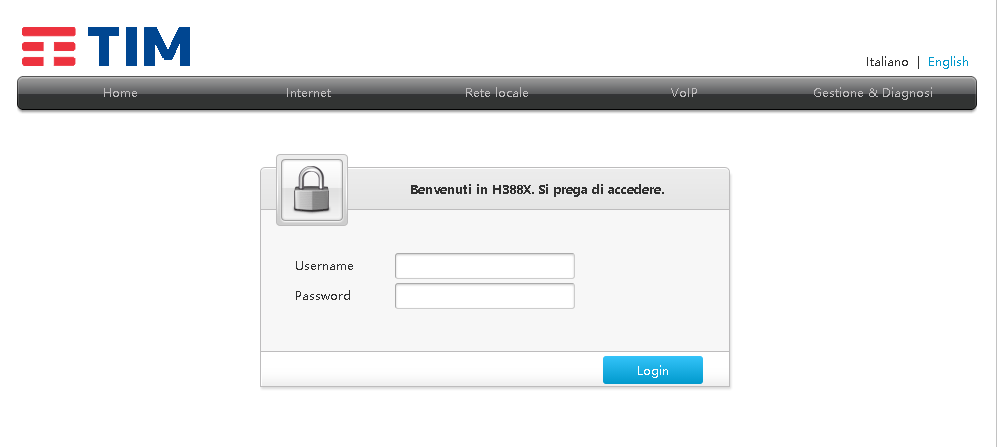
Per quanto riguarda gli agenti/rappresentanti, è molto fondamentale che abbiano strumenti che non richiedono conoscenze molto tecniche di informatica, per cui come sistema Operativo, anche per loro, gli verrà installato Windows 10 Home della Microsoft. Inoltre per i router (Figura 5) , abbiamo deciso facendo un'altra outsourcing, questa volta con la campagna telefonica e di internet, TIM, che ci procurerà tante SIM quanti sono i notebook disponibili.

Figura 5



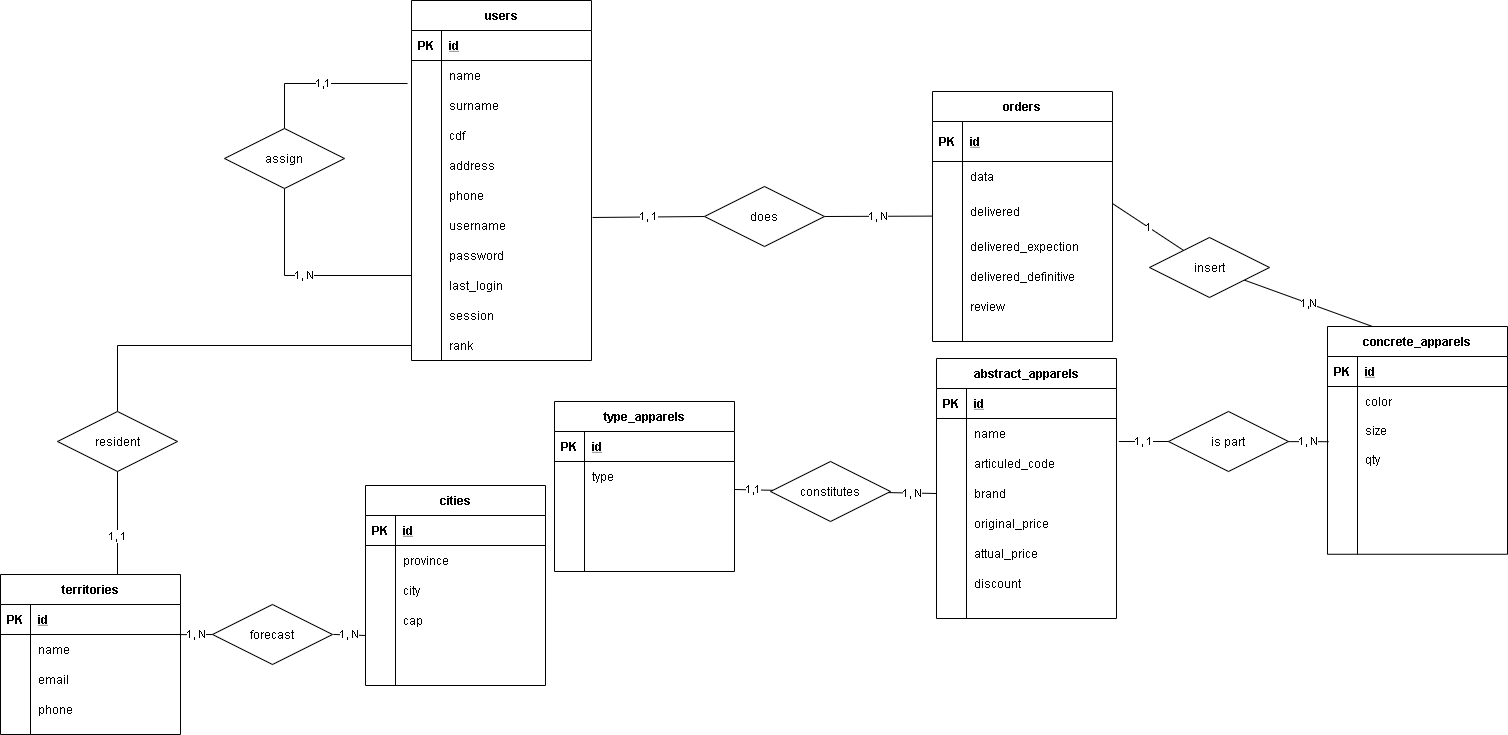
Tutti i notebook saranno predisposti con il firewall che negherà l’accesso da siti non opportuni e non inerenti all’ambiente di lavoro. Per far tutto ciò, ogni notebook saranno configurati in questo modo:  
Recarsi sull’indirizzo IP del nostro router, che per la TIM è 192.168.1.1 e verrà mostrata questa pagina e inserire le credenziali che verranno fornite direttamente dai tecnici della TIM tramite FAX all’azienda madre. Una volta entrati nell’index del nostro router, cliccare su ‘firewall’, cliccare nella voce ‘criteri filtro’ e tendina ‘filtro URL’ e verranno inseriti tutti i siti che reputiamo non produttive per l’agente. (Figura 6)

Figura 6



Database

Progettazione concettuale



Progettazione concettuale

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USERS | | | | |
| NAME | TYPE | ALLOW NULL | PRIMARY KEY | FOREIGN KEY |
| Id | Integer |  | X |  |
| Name | Varchar(20) |  |  |  |
| Surname | Varchar(20) |  |  |  |
| Cdf | Varchar(16) |  |  |  |
| Phone | Varchar(10) |  |  |  |
| Address | Varchar(50) |  |  |  |
| Territory | Integer |  |  | Territories(id) |
| Username | Varchar(50) | X |  |  |
| Password | Varchar(50) | X |  |  |
| Last\_login | Date | X |  |  |
| Session | Varchar(500) | X |  |  |
| Rank | Integer | X |  |  |
| Email | Varchar(100) | X |  |  |
| Custumer | Integer |  |  | Users(id) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TERRITORIES | | | | |
| NAME | TYPE | ALLOW NULL | PRIMARY KEY | FOREIGN KEY |
| Id | Integer |  | X |  |
| name | Varchar(30) |  |  |  |
| Email | Varchar(40) | X |  |  |
| phone | Varchar(10) |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CITIES | | | | |
| NAME | TYPE | ALLOW NULL | PRIMARY KEY | FOREIGN KEY |
| Id | Integer |  | X |  |
| province | Varchar(15) |  |  |  |
| city | Varchar(15) |  |  |  |
| cap | Varchar(5) |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CITIES\_TERRITORIES | | | | |
| NAME | TYPE | ALLOW NULL | PRIMARY KEY | FOREIGN KEY |
| Id\_territory | Integer |  | X | X |
| Id\_city | integer |  | X | X |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ORDERS | | | | |
| NAME | TYPE | ALLOW NULL | PRIMARY KEY | FOREIGN KEY |
| Id | Integer |  | X |  |
| Price | Double |  |  |  |
| Shipping | Date |  |  |  |
| Delivered | Date | X |  |  |
| review | Varchar(500) | X |  |  |
| Address | Varchar(50) |  |  |  |
| User | Integer |  |  | Users(id) |
| Apparel | Integer |  |  | Concrete\_apparels(id) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| APPARELS\_CONCRETE | | | | |
| NAME | TYPE | ALLOW NULL | PRIMARY KEY | FOREIGN KEY |
| Id | Integer |  | X |  |
| Color | Varchar(20) |  |  |  |
| size | Varchar(5) |  |  |  |
| apparel | integer |  |  | Abstract\_apparels(id) |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| APPARELS\_ABSTRACT | | | | |
| NAME | TYPE | ALLOW NULL | PRIMARY KEY | FOREIGN KEY |
| Id | Integer |  | X |  |
| Name | Varchar(20) |  |  |  |
| Articuled\_code | Varchar(15) |  |  |  |
| Brand | Varchar(30) |  |  |  |
| Original\_price | Double | X |  |  |
| Attual\_price | Double |  |  |  |
| discount | Double | X |  |  |
| Type | Integer |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| APPARELS\_TYPE | | | | |
| NAME | TYPE | ALLOW NULL | PRIMARY KEY | FOREIGN KEY |
| Id | Integer |  | X |  |
| type | Varchar(20) |  |  |  |
|  |  |  |  |  |

Progettazione fisica

-- ----------------------------

-- Table structure for `apparels\_abstract`

-- ----------------------------

DROP TABLE IF EXISTS `apparels\_abstract`;

CREATE TABLE `apparels\_abstract` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(20) NOT NULL,

`articuled\_code` varchar(15) NOT NULL,

`brand` varchar(30) NOT NULL,

`original\_price` double NOT NULL,

`attual\_price` double NOT NULL,

`discount` double NOT NULL,

`type` int(11) NOT NULL,

`url` varchar(255) NOT NULL,

PRIMARY KEY (`id`),

KEY `typeapp` (`type`),

CONSTRAINT `typeapp` FOREIGN KEY (`type`) REFERENCES `apparels\_type` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=latin1;

-- ----------------------------

-- Table structure for `apparels\_concrete`

-- ----------------------------

DROP TABLE IF EXISTS `apparels\_concrete`;

CREATE TABLE `apparels\_concrete` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`color` varchar(20) NOT NULL,

`size` varchar(5) NOT NULL,

`apparel` int(11) NOT NULL,

`hexcolor` varchar(10) NOT NULL DEFAULT '#fffff',

`qty` int(11) NOT NULL DEFAULT 50,

PRIMARY KEY (`id`),

KEY `concreabstr` (`apparel`),

CONSTRAINT `concreabstr` FOREIGN KEY (`apparel`) REFERENCES `apparels\_abstract` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=61 DEFAULT CHARSET=latin1;

-- ----------------------------

-- Table structure for `apparels\_type`

-- ----------------------------

DROP TABLE IF EXISTS `apparels\_type`;

CREATE TABLE `apparels\_type` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`type` varchar(20) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=latin1;

-- ----------------------------

-- Table structure for `cities`

-- ----------------------------

DROP TABLE IF EXISTS `cities`;

CREATE TABLE `cities` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`province` varchar(15) NOT NULL,

`city` varchar(15) NOT NULL,

`cap` varchar(5) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=16 DEFAULT CHARSET=latin1;

-- ----------------------------

-- Table structure for `logs`

-- ----------------------------

DROP TABLE IF EXISTS `logs`;

CREATE TABLE `logs` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`id\_user` int(11) DEFAULT NULL,

`action` varchar(500) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `usrlog` (`id\_user`),

CONSTRAINT `usrlog` FOREIGN KEY (`id\_user`) REFERENCES `users` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=33 DEFAULT CHARSET=latin1;

-- ----------------------------

-- Table structure for `orders`

-- ----------------------------

DROP TABLE IF EXISTS `orders`;

CREATE TABLE `orders` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`price` double NOT NULL,

`shipping` date NOT NULL,

`delivered` date NOT NULL,

`review` varchar(500) NOT NULL,

`address` varchar(50) NOT NULL,

`user` int(11) NOT NULL,

`apparel` int(11) NOT NULL,

PRIMARY KEY (`id`),

KEY `userorder` (`user`),

CONSTRAINT `userorder` FOREIGN KEY (`user`) REFERENCES `users` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=latin1;

-- ----------------------------

-- Table structure for `territories`

-- ----------------------------

DROP TABLE IF EXISTS `territories`;

CREATE TABLE `territories` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(30) NOT NULL,

`email` varchar(40) NOT NULL,

`phone` varchar(10) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=latin1;

-- ---------------------------

-- Table structure for `territories\_cities`

-- ----------------------------

DROP TABLE IF EXISTS `territories\_cities`;

CREATE TABLE `territories\_cities` (

`territory` int(11) NOT NULL,

`city` int(11) NOT NULL,

PRIMARY KEY (`city`,`territory`),

KEY `tercity` (`territory`),

CONSTRAINT `cityter` FOREIGN KEY (`city`) REFERENCES `cities` (`id`),

CONSTRAINT `tercity` FOREIGN KEY (`territory`) REFERENCES `territories` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- ----------------------------

-- Table structure for `users`

-- ----------------------------

DROP TABLE IF EXISTS `users`;

CREATE TABLE `users` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(20) DEFAULT NULL,

`surname` varchar(20) DEFAULT NULL,

`cdf` varchar(16) DEFAULT NULL,

`phone` varchar(10) DEFAULT NULL,

`territory` int(11) DEFAULT NULL,

`username` varchar(50) DEFAULT NULL,

`password` varchar(50) DEFAULT NULL,

`last\_login` date DEFAULT NULL,

`rank` int(11) DEFAULT 1,

`email` varchar(100) DEFAULT NULL,

`custumer` int(11) DEFAULT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `cdfunq` (`cdf`) USING BTREE,

KEY `cust` (`custumer`),

KEY `terruser` (`territory`),

CONSTRAINT `cust` FOREIGN KEY (`custumer`) REFERENCES `users` (`id`),

CONSTRAINT `terruser` FOREIGN KEY (`territory`) REFERENCES `territories` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=41 DEFAULT CHARSET=latin1;