



Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI CIVITA"

Corso di laurea in INFORMATICA

**Gestione delle risorse aziendali
mediante un sistema ERP in ambito
Food and beverage**

Relatore

DOTT. ARMIR BUJARI
UNIVERSITÀ DI PADOVA

Laureando

GIOVANNI BERGO

Alla Mia Famiglia.

Ringraziamenti

Un altro traguardo molto importante è stato raggiunto e sono molto felice di condividerlo con coloro che mi sono stati vicino durante questi anni.

Prima di tutto vorrei porgere i miei ringraziamenti al Professor Armir Bujari per il supporto fornитomi durante lo stage e la stesura della tesi.

Ai miei compagni di Università, per tutte le giornate passate insieme a studiare e le esperienze affrontate.

Infine, ringrazio i miei genitori, per darmi un futuro migliore, ma per i quali le parole non sono sufficienti.

Padova, Settembre 2019

Giovanni Bergo

Sommario

Il documento illustra il lavoro svolto durante lo stage curriculare, della durata di trecentoventi ore, presso l'azienda Sync Lab.

Lo scopo di tale stage è stato quello di gestire le risorse di un'azienda mediante un sistema ERP(Odoo) in ambito Food-and-beverage, in pratica aggiungere funzionalità aggiuntive ai moduli Odoo. Sono stati eseguiti dei test per verificare che il rispettivo codice sviluppato sia corretto e funzionante.

Alcune delle caratteristiche in cui si è andati a lavorare sono: la gestione di vendite e acquisti, la gestione del magazzino e la gestione delle risorse umane.

Gli obiettivi di maggiore importanza sono stati l'apprendimento del linguaggio di programmazione Python e delle librerie correlate, utili per la creazione del modello così da poter manipolare i dati ed eseguire test di unità su di essi; l'approfondimento del linguaggio Xml/XPath per la creazione delle views.

Indice

1	INTRODUZIONE	1
1.1	L'azienda	1
1.2	L'idea	2
1.3	Organizzazione del testo	2
2	PIANO DI LAVORO	5
2.1	Contesto	5
2.2	Introduzione al progetto	6
2.3	Vincoli temporali, tecnologici e metodologici	6
2.4	Requisiti e obiettivi	7
2.5	Pianificazione	8
2.6	Ambiente di lavoro	10
2.6.1	Metodi di sviluppo	10
2.6.2	Gestione di progetto	10
2.6.3	Linguaggio di programmazione e ambiente di sviluppo . .	10
2.7	Analisi dei rischi	11
3	TECNOLOGIE E STRUMENTI	13
3.1	Tecnologie	13
3.1.1	Python	13
3.1.2	ORM Api	14
3.1.3	XML/XPATH	15
3.2	Strumenti	15
3.2.1	OpenProject	15
3.2.2	PostgreSQL	16
3.2.3	PyCharm	16
3.2.4	pgAdmin 4	17
3.2.5	Homebrew	17
3.2.6	PIP	18
3.2.7	Odoo	18
4	MODULI FOOD-AND-BEVERAGE	19
4.1	Modulo odoo standard	19
4.1.1	Composizione di un modulo	19
4.1.2	Struttura di un modulo	20

4.2	Moduli sviluppati	23
4.2.1	Limite di credito del partner	24
4.2.2	Canali di Vendita	28
4.2.3	Gestione imballi	30
5	TEST E VALIDAZIONE	33
5.1	Test di sistema e collaudo	33
5.1.1	Limite di credito del partner	33
5.1.2	Canali di Vendita	39
5.1.3	Gestione imballi	39
5.2	Test di Unità	45
6	CONCLUSIONI	47
6.1	Consuntivo finale	47
6.2	Raggiungimento degli obiettivi	48
6.3	Conoscenze acquisite	49
6.4	Sviluppi futuri del software	49
6.5	Valutazione personale	50
GLOSSARIO		51
ACRONIMI		52
RIFERIMENTI BIBLIOGRAFICI		55

Lista delle figure

1.1	Logo Sync Lab	1
3.1	Logo Python	13
3.2	ORM API	14
3.3	Logo OpenProject	15
3.4	Logo Postgresql	16
3.5	Logo PyCharm	16
3.6	Logo pgAdmin 4	17
3.7	Logo Homebrew	17
3.8	Logo Pip	18
3.9	Logo Odoo	18
4.1	Struttura di un modulo	20
4.2	File manifest	21
4.3	Comando per configurare un nuovo modulo	21
4.4	Modelli creati	22
4.5	Diagramma moduli sviluppati	23
4.6	Popup limite di credito	25
4.7	Popup blocco scaduto	27
4.8	Inserimento nuovo cliente: Persona	28
4.9	Inserimento nuovo cliente: Azienda	29
4.10	Inserimento pacchetti di prodotti	30
4.11	Campo Pianali e Automezzo in Ordini di vendita	31
4.12	Campi checkbox in Prodotti	32
5.1	Test senza errori	34
5.2	Esempio limite di credito	35
5.3	Test limite di credito	35
5.4	Prezzo di listino settato	36
5.5	Popup d'errore sul prezzo unitario	37
5.6	Esempio blocco scaduto	38
5.7	Test blocco scaduto	38
5.8	Test canale di vendita	39
5.9	Esempio form confezione	40
5.10	Primo Test campo 'Pianali'	41

5.11	Secondo Test campo 'Pianali'	42
5.12	Terzo Test campo 'Pianali'	43
5.13	Inserimento di uno stesso prodotto	44
5.14	Test sulle <i>line</i> dei prodotti	44
5.15	Test di unità	45

Lista delle tabelle

2.1 Pianificazione concordata nel piano di lavoro	9
6.1 Ripartizione reale delle ore di stage	48

1

Introduzione

1.1 L'azienda



Figure 1.1: Logo Sync Lab

Sync Lab è un'azienda che nasce come Software house tramutatasi rapidamente in una società di consulenza informatica attraverso un processo di maturazione delle competenze tecnologiche, e dal 2002 ad oggi ha raggiunto un organico aziendale di oltre 200 risorse. L'azienda, propone sul mercato interessanti quanto innovativi prodotti software, nati nel proprio laboratorio di ricerca e sviluppo.

L'azienda ha sede legale a Napoli, ma ha anche altre sedi, come: Padova, Milano, Roma. Ha gradualmente conquistato significativamente fette di mercato nei seguenti settori: mobile, videosorveglianza e sicurezza delle infrastrutture informatiche aziendali.

Da questa realtà si evince come si debba essere sempre pronti a supportare il cliente nella realizzazione, messa in Opera e governance di soluzioni IT, sia dal punto di vista tecnologico, sia nel governo del cambiamento organizzativo.

1.2 L'idea

Il progetto che sta realizzando l'azienda, è in via di sviluppo e si basa sulla gestione delle risorse di un'azienda in ambito *Food-and-beverage^[g]*. L'idea è quella di mantenere ottimali i livelli di inventario riguardanti il *ciclo di vita^[g]* degli alimenti e delle bevande, valutandoli tramite test per verificarne la correttezza del modulo ed il funzionamento dello stesso.

1.3 Organizzazione del testo

Di seguito viene riportata per ogni capitolo una piccola descrizione delle tematiche trattate:

- **Capitolo 2:** in questo capitolo vengono riportati gli obiettivi, il progetto in dettaglio e la pianificazione concordata con l'azienda, inoltre vengono riportate le metodologie e strumenti utilizzati in generale, infine un'analisi dei rischi;
- **Capitolo 3:** vengono riportati gli strumenti adottati per lo svolgimento delle attività, corredati da una breve descrizione che riporti come sono stati utilizzati;
- **Capitolo 4:** viene riportata la descrizione di un modulo standard Odoo, i moduli sviluppati nel progetto Food-and-beverage con una descrizione delle loro caratteristiche e la lista delle funzionalità sviluppate; inoltre ci sarà del materiale grafico utilizzato durante lo stage;
- **Capitolo 5:** vengono riportate le modalità con cui si sono eseguiti i test, illustrando come siano stati strutturati e come sia stato possibile verificare le soluzioni fornite dai moduli;
- **Capitolo 6:** vengono riportate le conclusioni relative al numero di obiettivi soddisfatti e all'effettiva suddivisione delle ore rispetto tali obiettivi.

Il testo adotta le seguenti convenzioni tipografiche:

- ogni acronimo, abbreviazione, parola ambigua o tecnica viene spiegata e chiarificata alla fine del testo;
- ogni parola di glossario alla prima apparizione verrà etichettata come segue: *parola^[g]*;
- ogni riga di un elenco puntato terminerà con un ; a parte l'ultima riga che si concluderà con un punto.

2

Piano di lavoro

In questo capitolo verranno riportati in modo approfondito lo scopo e gli obiettivi dello stage, contestualizzazione delle attività alla realtà aziendale e scadenzario delle stesse.

2.1 Contesto

Il progetto generale nasce dalla visione dell’azienda, di studiare e implementare moduli sulla piattaforma ERP open Odoo.

Il software è *open source^[g]* e si può considerare una suite di prodotti software attivamente supportata da una community internazionale ed una italiana ai fini della localizzazione. Alcune caratteristiche che permettono di soddisfare gli obiettivi sono: contabilità finanziaria, contabilità analitica, gestione del magazzino, gestione di vendite e acquisti, automazione dei processi, gestione risorse umane. (altre caratteristiche della piattaforma: campagne di marketing, siti web aziendali ed e-commerce).

2.2 Introduzione al progetto

Il progetto di stage, dopo l'evento StageIt¹³ ed alcune riunioni presso l'azienda, è stato studiato dettagliatamente e riportato nel documento "piano di lavoro", nello stesso sono riportati gli obiettivi e la pianificazione delle attività.

In particolare, il progetto fa riferimento a problematiche in ambito produttivo "Food and beverage", realizzando un modulo che si occupa di monitorare il ciclo di vita degli alimenti e delle bevande, il tutto per ottimizzare il cosiddetto *Time-to-Market*^[g], mantenendo ottimali i livelli di inventario e ottimizzando i flussi economici dei prodotti. Lo stage prevede la realizzazione di diversi moduli il cui sviluppo sarà *incrementale*^[g].

A rendere più complicato il lavoro dell'apprendimento di Odoo riportiamo alcuni principali problemi da tenere in considerazione:

- Gestione del database: si appoggia a pgAdmin 4⁷, che consente di gestire in modo semplificato database PostgreSQL, verificando il corretto inserimento o eliminazione di dati;
- Gestire gli errori: organizzare una buona vista tramite i tag Odoo, XML e XPath idonei, in modo da controllare gli errori e avvisi.

2.3 Vincoli temporali, tecnologici e metodologici

Nel periodo di stage svolto presso l'azienda mi è stato chiesto di svolgere dei task e tenere traccia del loro stato di completamento, tramite un applicazione via web condivisa, OpenProject⁵, nel suddetto mi si richiedeva di annotare giornalmente l'avanzare del lavoro riportando idee e osservazioni emerse durante i *brainstorming*^[g] quotidiani, criticità rilevate e positività riscontrate negli strumenti utilizzati. Le telefonate con i ragazzi delle sede di Napoli, che seguivano il progetto, sono state molto importanti per chiarire le consegne date su ogni task e la corretta organizzazione del codice da implementare.

Prima di iniziare lo stage è stato concordato con l'azienda un piano di lavoro su un totale di 320 ore, lavorando 5 giorni a settimana, 8 ore per ciascun giorno.

2.4 Requisiti e obiettivi

Nell'elenco di seguito vengono riportati gli obiettivi dello stage, a corredo degli stessi vi sarà un codice univoco ed una breve descrizione.

Ogni obiettivo è provvisto di un codice identificativo formato da una delle seguenti stringhe *ob,de,op*, che rappresentano il livello di importanza e da un numero incrementale positivo, che rispetta la seguente nomenclatura:

[importanza][identificativo].

Il livello di importanza di ciascun obiettivo può essere uno tra i seguenti:

- Obbligatorio: individuato dalla stringa *ob*, sono obiettivi fondamentali per la riuscita del progetto, il loro soddisfacimento dovrà verificarsi obbligatoriamente entro la fine dello stage, pena il fallimento dello stesso;
- Desiderabile: individuato dalla stringa *de*, sono obiettivi secondari su cui però si nutre dell'interesse, il loro soddisfacimento è auspicabile entro la fine dello stage;
- Opzionale: individuato dalla stringa *op*, sono obiettivi di contorno su cui si nutre poco interesse, la loro realizzazione si verificherà nel momento in cui si dovessero soddisfare tutti gli obiettivi obbligatori e desiderabili prima della fine dello stage.

Di seguito riportiamo la lista degli obiettivi pianificati:

- Obbligatori
 - *ob01*: Implementazione di un modulo che si occuperà di monitorare il ciclo di vita degli alimenti e delle bevande con l'obiettivo di ottimizzare il cosiddetto Time-to-Market, mantenendo ottimali i livelli di inventario ed ottimizzando i flussi economici dei prodotti;
 - *ob02*: Acquisizione competenze sulle tematiche sopra descritte;
 - *ob03*: Capacità di raggiungere gli obiettivi richiesti in autonomia seguendo il programma;
 - *ob04*: Portare a termine le modifiche richieste dal cliente con una percentuale di superamento degli item di collaudo pari al 50%.

- Desiderabili
 - *de01*: Acquisizione di un buon livello di autonomia sulla piattaforma Erp Odoo;
 - *de02*: Portare a termine le modifiche richieste dal cliente con una percentuale di superamento degli item di collaudo pari all'80%.
- Opzionali
 - *op01*: Implementazione di nuovi moduli.

2.5 Pianificazione

Con le ore a disposizione per questo stage si è proceduto a organizzare come segue le attività:

- **Formazione**: si è visto necessario approfondire i database PostgreSQL⁹, il linguaggio di markup XML-XPath, Html5/CSS3, il linguaggio di scripting orientato agli oggetti e agli eventi Javascript³, imparare il linguaggio di programmazione Python¹¹ e lo studio della Piattaforma Erp Odoo⁴
- **Sviluppo moduli Odoo**: Implementazione nuovi moduli in ambito Food-and-Beverage o modifiche su esistenti
- **Collaudo Finale**: Esecuzione dei test e collaudo

La pianificazione delle attività è stata distribuita come mostrato nella Tabella 2.1.

Durata in ore		Descrizione dell'attività
208		A: Formazione
	15	• Studio Database Sql-PostgreSQL
	15	• Studio XML-XPath
	10	• Studio Html5/CSS3
	20	• Studio Javascript
	53	• Studio linguaggio Python
	95	• Studio Piattaforma Erp Odoo
72		B: Sviluppo moduli
	72	• Implementazione di nuovi moduli Odoo o modifiche su esistenti
40		C: Collaudo Finale
	20	• Test
	15	• Stesura Documentazione Finale
	5	• Collaudo e consegna del codice
Totale: 320		

Table 2.1: Pianificazione concordata nel piano di lavoro

2.6 Ambiente di lavoro

2.6.1 Metodi di sviluppo

Il *ciclo di vita*^[g] di un prodotto Sync Lab segue il *modello evolutivo*^[g], formato da un periodo di concezione dell'idea, analisi della stessa, realizzazione di una singola evoluzione ed infine, si rilascia il prodotto, che può avere più versioni esterne attive in parallelo e dovrà mostrare le nuove funzionalità sviluppate dimostrando così l'incremento fatto. In quest'ottica, lo sviluppo di un modulo può essere associato ad un incremento, la cui *milestone*^[g] è il passaggio dei test.

2.6.2 Gestione di progetto

Per quanto riguarda la gestione di progetto sono stati utilizzati alcuni strumenti descritti con maggiore dettaglio nel **Capitolo 3**. In generale per la gestione dei task da eseguire si è fatto uso di OpenProject⁵, uno strumento di *project management*^[g]; per la gestione della comunicazione e informazioni si è fatto uso dell'applicazione Slack¹² e per la condivisione di documentazione e articoli si è fatto uso della e-mail.

2.6.3 Linguaggio di programmazione e ambiente di sviluppo

Per la totalità dello stage si è lavorato utilizzando PyCharm¹⁰. Con questo strumento è stato possibile scrivere il modello in linguaggio Python¹¹ in modo molto agevole.

Questo linguaggio di programmazione è orientato agli oggetti ed interpretato dinamicamente al momento dell'esecuzione da un interprete. Python¹¹ risulta veramente versatile in quanto fornisce incredibili funzionalità utilizzabili in modo semplice e intuitivo, dispone di moltissimi moduli che permettono le più svariate operazioni.

2.7 Analisi dei rischi

In questa sezione vengono riportati i principali rischi che si prefiguravano all'inizio dello stage. Ciascuno di essi oltre ad avere una breve descrizione riporta il livello di rischio, in termini di pericolosità per la riuscita dello stage e come si possa fare per evitarlo:

- **Difficoltà nelle tecnologie adottate**

Ad inizio stage è stato chiaro che Python¹¹ avrebbe avuto un ruolo dominante nel progetto, la mole di librerie ed estensioni rendeva il linguaggio troppo vasto da poter approfondire nella sua interezza, ed oltre a questo vi erano le views da imparare

- **Livello di rischio:** Medio;
- **Contromisure:** Studiare approfonditamente la costruzione di un modello Odoo tramite esercitazione.

- **Difficoltà di integrazione nel team**

Di fondamentale importanza per la riuscita di un progetto è la cooperazione con i colleghi e la creazione di un ambiente di lavoro sano e che stimoli la produttività, essendo un nuovo arrivato inserito in un ambiente soggetto a forte stress per le stringenti scadenze vi era la possibilità di entrare in conflitto con qualche collega.

- **Livello di rischio:** Basso;
- **Contromisure:** Perseguire un atteggiamento positivo, critico e oggettivo.

- **Contrattempi dovuti a malattie e impegni**

Un rischio da tenere in considerazione è quello dovuto a impegni o malattie che precludano la possibilità di recarsi nel luogo di lavoro, data la durata dello stage è sicuramente possibile possa verificarsi.

- **Livello di rischio:** Basso;
- **Contromisure:** Organizzare precedentemente ogni impegno non lavorativo e tempo di *slack*^[g] per evitare contrattempi.

- **Difficoltà di stima dei tempi previsti**

Con un progetto di così lunga durata e l'inesperienza che ci portiamo appresso è possibile che vengano fatti degli errori di valutazione in termini di tempistiche per lo svolgimento delle diverse attività pianificate.

- **Livello di rischio:** Medio;
- **Contromisure:** Rendere partecipi nella definizione del piano di lavoro persone esperte, come il tutor aziendale e il project manager.

3

Tecnologie e strumenti

Questo capitolo riporta le tecnologie e gli strumenti utilizzati durante il corso dello stage.

3.1 Tecnologie

Le tecnologie studiate sono:

- Python;
- ORM Api;
- Xml/XPath.

3.1.1 Python



Figure 3.1: Logo Python

Python¹¹, il cui logo riportato in Figura 3.1, è un linguaggio di programmazione tra i più famosi attualmente disponibili, è un linguaggio ad oggetti, sintatticamente semplice e intuitivo che permette di avere un approccio ai problemi molto più operativo. Python 3.7.0 è la versione e la base tecnologica su cui si è basato lo stage e gli strumenti utilizzati.

3.1.2 ORM Api



Figure 3.2: ORM API

ORM Api⁶ fornisce, mediante un’interfaccia orientata agli oggetti, tutti i servizi inerenti alla persistenza dei dati, astraendo nel contempo le caratteristiche implementative dello specifico RDBMS utilizzato.

I principali vantaggi nell’uso di un tale sistema sono i seguenti:

- Un’elevata portabilità rispetto alla tecnologia DBMS utilizzata: cambiando DBMS non devono essere riscritte le routine che implementano lo strato di persistenza;
- Drastica riduzione della quantità di codice sorgente da redigere; l’ORM maschera dietro semplici comandi le complesse attività di creazione, prelievo, aggiornamento ed eliminazione dei dati. Tali attività occupano di solito una buona percentuale del tempo di stesura, testing e manutenzione complessivo;
- Suggerisce la realizzazione dell’architettura di un sistema software mediante approccio stratificato, tendendo pertanto ad isolare in un solo livello la logica di persistenza dei dati, a vantaggio della modularità complessiva del sistema.

L’uso di un ORM favorisce il raggiungimento di più alti standard qualitativi software, migliorando in particolare le caratteristiche di correttezza, manutenibilità, evolvibilità e portabilità.

3.1.3 XML/XPATH

XML/ XPath è un metalinguaggio per la definizione di linguaggi di markup, ovvero un linguaggio marcatore basato su un meccanismo sintattico che consente di definire e controllare il significato degli elementi contenuti in un documento o in un testo.

3.2 Strumenti

Gli strumenti utilizzati sono:

- OpenProject;
- PostgreSQL;
- PyCharm;
- pgAdmin;
- Homebrew;
- PIP;
- Odoo.

3.2.1 OpenProject



Figure 3.3: Logo OpenProject

OpenProject⁵, il cui logo riportato in Figura 3.3, è una piattaforma di project management utilizzata durante il progetto per la gestione delle task da soddisfare in un certo periodo di tempo e l'aggiornamento del loro stato di esecuzione.

3.2.2 PostgreSQL

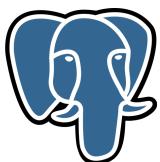


Figure 3.4: Logo Postgresql

PostgreSQL⁹, il cui logo riportato in Figura 3.6, è un completo DBMS ad oggetti. Odoo ha bisogno di un server PostgreSQL per funzionare correttamente. Al termine dell'installazione del software richiesto, è necessario creare un utente PostgreSQL. Odoo utilizzerà questo utente per connettersi a PostgreSQL.

3.2.3 PyCharm



Figure 3.5: Logo PyCharm

PyCharm¹⁰, il cui logo riportato in Figura 3.5, è un IDE utilizzato in particolare per il linguaggio Python¹¹. Fornisce analisi del codice, un debugger grafico, un tester di unità integrato, integrazione con i sistemi di controllo di versione e supporta lo sviluppo web con Django e Data Science con Anaconda. PyCharm è multipiattaforma , con versioni Windows , macOS e Linux .

3.2.4 pgAdmin 4

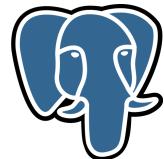


Figure 3.6: Logo pgAdmin 4

pgAdmin 4⁷ è un'applicazione che consente di amministrare in modo semplificato database PostgreSQL. L'applicazione è indirizzata sia agli amministratori del database, sia agli utenti. Gestisce i permessi prelevandoli dal database PostgreSQL. E' stato uno strumento molto utile durante lo stage perchè mi ha permesso di monitorare costantemente il database da qualsiasi aggiornamento, grazie alla sua semplicità, verificandone il corretto salvataggio dei dati su ogni modulo.

3.2.5 Homebrew



Figure 3.7: Logo Homebrew

Homebrew², il cui logo riportato in Figura 3.7, è un gestore di pacchetti che semplifica l'installazione di software sul sistema operativo macOS di Apple e Linux. Fondamentale per l'installazione di pacchetti non presenti nella libreria standard, di facile utilizzo ha permesso di installare velocemente pacchetti indispensabili per il proseguimento dello stage.

3.2.6 PIP



Figure 3.8: Logo PIP

PIP⁸, il cui logo riportato in Figura 3.8, è un gestore di pacchetti fondamentale per l’installazione delle dipendenze Python.

3.2.7 Odoo



Figure 3.9: Logo Odoo

Odoo⁴, il cui logo riportato in Figura 3.9, è un software di gestione aziendale a struttura modulare. Ogni modulo risponde alle esigenze di informatizzazione di un’area funzionale dell’impresa, ecco le principali:

- Moduli per creare il sito web aziendale (Website Builder, e-Commerce, Blogs);
- Moduli inerenti il ciclo delle vendite (CRM, Preventivi, Punto Vendita);
- Moduli per la gestione aziendale (Gestione Progetti, Fatturazione, Contabilità, Magazzino, Produzione, Acquisti);
- Moduli di marketing (Campagne eMails, Eventi, Automatizzazione delle tentate vendite).

4

Moduli Food-and-beverage

In questo capitolo presenteremo prima una rapida introduzione ai moduli odoo, tratta dalla documentazione¹ degli sviluppatori, per poi analizzare nel dettaglio ciascun modulo implementato, spiegando ciascun vincolo e delineandone quindi l'idea di base.

4.1 Modulo odoo standard

Odoo usa un'architettura client/server nella quale i client sono web browser che accedo al server odoo tramite *RPC^[g]*. La logica aziendale e l'estensione vengono generalmente eseguite sul lato server. Entrambe le estensioni server e client sono impacchettate come moduli che possono essere caricati facoltativamente in un database. I moduli Odoo possono sia aggiungere una nuovissima logica aziendale a un sistema Odoo, sia alterare ed estendere la logica aziendale esistente.

Tutto in Odoo quindi inizia e termina con i moduli.

4.1.1 Composizione di un modulo

Un modulo Odoo può contenere un numero di elementi:

- **Oggetti business** Dichiarate come classi Python, queste risorse vengono automaticamente mantenute da Odoo in base alla loro configurazione
- **File di dati** File XML o CSV che dichiarano metadati (viste o report), dati di configurazione (parametrizzazione dei moduli), dati dimostrativi e altro
- **Controller Web** Gestire le richieste dai browser Web
- **Dati web statici** Immagini, file CSS o javascript utilizzati dall’interfaccia Web o dal sito Web

4.1.2 Struttura di un modulo

Ogni modulo è una directory all’interno di una directory contenente tutti i moduli Odoo, specificata utilizzando il comando ”- -addons-path <nome_modulo>” nel file di configurazione.

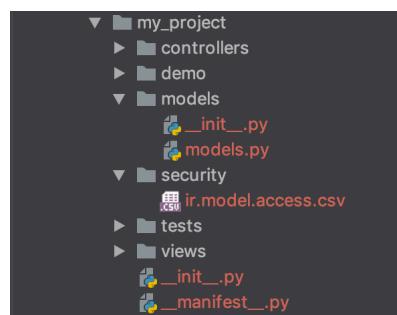


Figure 4.1: Struttura modulo

Un modulo è dichiarato nel suo file ”__manifest__.py”, per specificare i metadati del modulo, e anche su un file Python ”init.py”, contenente l’importazione del controller e modello.

The screenshot shows a code editor window with the title bar "manifest.py" and status bar "Line 1, Column 1" and "Python". The code is a Python dictionary defining a module manifest. It includes fields for name, summary, description, author, website, category, version, depends, data, and demo.

```
# -*- coding: utf-8 -*-
{
    'name': "my_project",
    'summary': """
        Short (1 phrase/line) summary of the module's purpose, used as
        subtitle on modules listing or apps.openerp.com""",
    'description': """
        Long description of module's purpose
        """,
    'author': "My Company",
    'website': "http://www.yourcompany.com",
    # Categories can be used to filter modules in modules listing
    # Check https://github.com/odoo/odoo/blob/11.0/odoo/addons/base/module/module_data.xml
    # for the full list
    'category': 'Uncategorized',
    'version': '0.1',
    # any module necessary for this one to work correctly
    'depends': ['base', 'partner_credit_limit'],
    # always loaded
    'data': [
        # 'security/ir.model.access.csv',
        'views/views.xml',
        'views/templates.xml',
        'views/mail.xml'
    ],
    # only loaded in demonstration mode
    'demo': [
        'demo/demo.xml',
    ],
}
```

Figure 4.2: File manifest

Odoo fornisce un comando per configurare un nuovo modulo (vuoto), creando automaticamente un gruppo di file standard. La maggior parte di essi contiene semplicemente codice commentato o XML:

```
$ odoo-bin scaffold <nome modulo> <dove inserirlo>
```

Figure 4.3: Comando per configurare un nuovo modulo

Ogni modello odoo è identificato con un id (univoco) ed un nome che può essere individuato nelle impostazioni di odoo. Qui potremo individuare tutte le informazioni riguardo un dato modello, ad esempio: chi eredita, che campi contiene, autorizzazioni di accesso e viste.

Tutti questi dati permettono oggettivamente di valutare la corretta implementazione e creazione del modello.

Modello	Descrizione Modello	Tipo	Modello Transitorio
crm.team	Canale di Vendita	Oggetto Base	<input type="checkbox"/>
mail.channel	Canale di discussione	Oggetto Base	<input type="checkbox"/>
res.partner	Contatto	Oggetto Base	<input type="checkbox"/>
account.analytic.account	Conto analitico	Oggetto Base	<input type="checkbox"/>
account.bank.statement	Estratto conto bancario	Oggetto Base	<input type="checkbox"/>
calendar.event	Evento	Oggetto Base	<input type="checkbox"/>
account.invoice	Fattura	Oggetto Base	<input type="checkbox"/>
stock.production.lot	Lotto/Seriale	Oggetto Base	<input type="checkbox"/>
mail.test	Modello Test per Messaggi	Oggetto Base	<input type="checkbox"/>
product.template	Modello prodotto	Oggetto Base	<input type="checkbox"/>
account.payment	Pagamenti	Oggetto Base	<input type="checkbox"/>
sale.order	Preventivo	Oggetto Base	<input type="checkbox"/>
product.product	Prodotto	Oggetto Base	<input type="checkbox"/>
mail.test.simple	Provare un semplice record Chatter	Oggetto Base	<input type="checkbox"/>
mail.thread	Thread Email	Oggetto Base	<input type="checkbox"/>
stock.picking	Trasferimento	Oggetto Base	<input type="checkbox"/>

Figure 4.4: Modelli creati

4.2 Moduli sviluppati

I moduli estesi e modificati sono:

- Limite di credito del partner (my_project);
- Canali di Vendita (ibg_code);
- Gestione imballi (ibg_packaging).

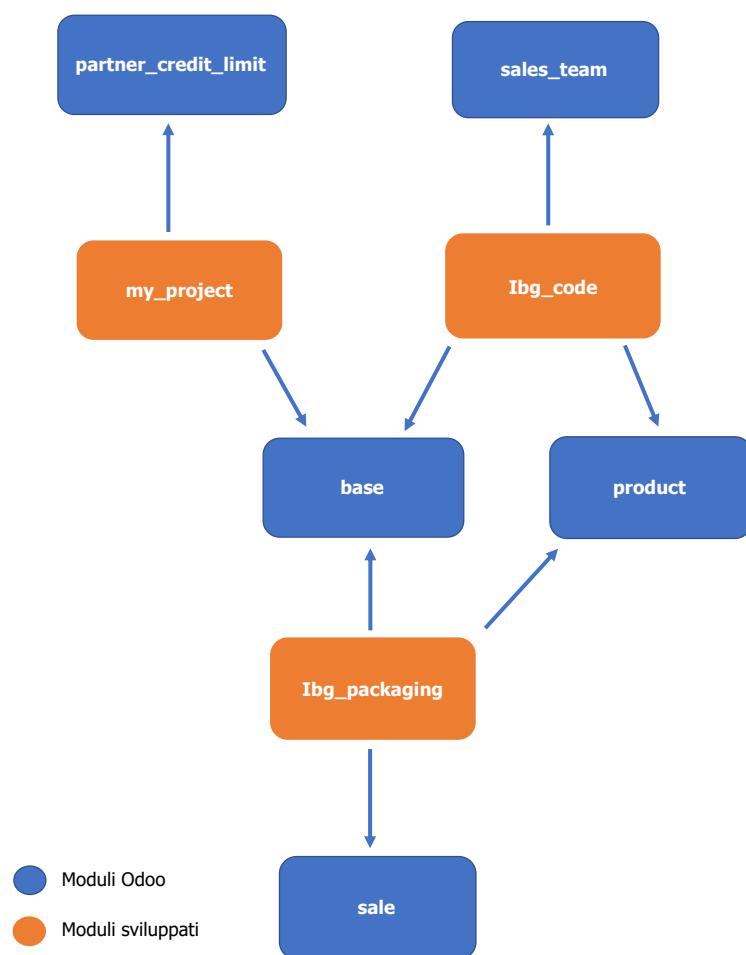


Figure 4.5: Diagramma moduli sviluppati

4.2.1 Limite di credito del partner

Questo modulo entra in funzione quando si approva un ordine di vendita, calcola la somma di:

- Il credito che il partner deve pagare;
- L'importo degli ordini di vendita approvati ma non ancora fatturati;
- L'importo dell'ordine di vendita da approvare e lo confronta con il limite di credito del partner. Se il limite di credito è inferiore, non consente di approvare l'ordine di vendita;
- Se l'opzione Consenti credito è selezionata, il sistema non controllerà i limiti di credito e consentirà a quel partner di ignorare il limite.

Qui abbiamo aggiunto le seguenti funzionalità.

Inseriamo gli stati Blocco Scaduto e Limite di credito.

Quando raggiungiamo lo stato Limite di Credito, stiamo confermando un ordine con un importo totale superiore al credito del partner.

Al momento della conferma d'ordine verrà mostrato un popup contenente:

- Cliente;
- Il credito disponibile del partner.

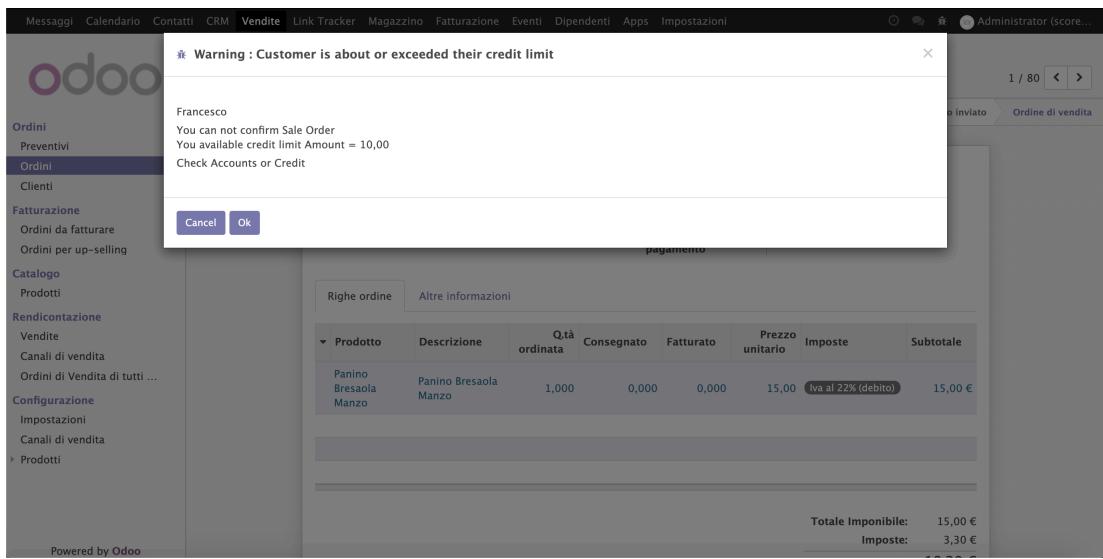


Figure 4.6: Popup limite di credito

Al momento del click sul tasto ok, del popup, la fattura si sposterà nello stato 'Limite di Credito', dove non sarà possibile modificarla, e verrà inviata una mail al direttore commerciale con scritto:

Caro Administrator,

l'ordine di vendita n° SO del cliente nome cliente è stato bloccato per Limite di Credito.

Puoi accedere al SO dal seguente link: link al SO di riferimento.

Inoltre è stato aggiunto un controllo al momento dell'inserimento di un prodotto. Se cerchiamo di modificare il prezzo unitario (>, < del prezzo di listino) dall'interfaccia 'Righe ordine', salviamo e confermiamo l'ordine, si visualizzerà un popup d'errore con scritto:

Prezzo non idoneo !

Quando raggiungiamo lo stato Blocco Scaduto, stiamo confermando un ordine con una data di scadenza antecedente a quella attuale, vuol dire che l'ordine di vendita è già scaduto.

Al momento della conferma d'ordine verrà mostrato un popup contenente:

- Cliente;
- Fatture Scadute: campo contenente l'elenco delle fatture scadute del cliente;
- Importo dello scaduto: campo contenente la somma del totale delle fatture scadute;
- Button OK

Inoltre al momento del click del tasto ok del popup, la fattura si sposterà nello stato 'Blocco Scaduto', dove non sarà possibile modificarla, e verrà inviata una mail al direttore commerciale con scritto:

*Caro Administrator,
l'ordine di vendita n° SO del cliente nome cliente è stato bloccato per presenza di pagamenti scaduti.
Puoi accedere al SO dal seguente link: link al SO di riferimento.*

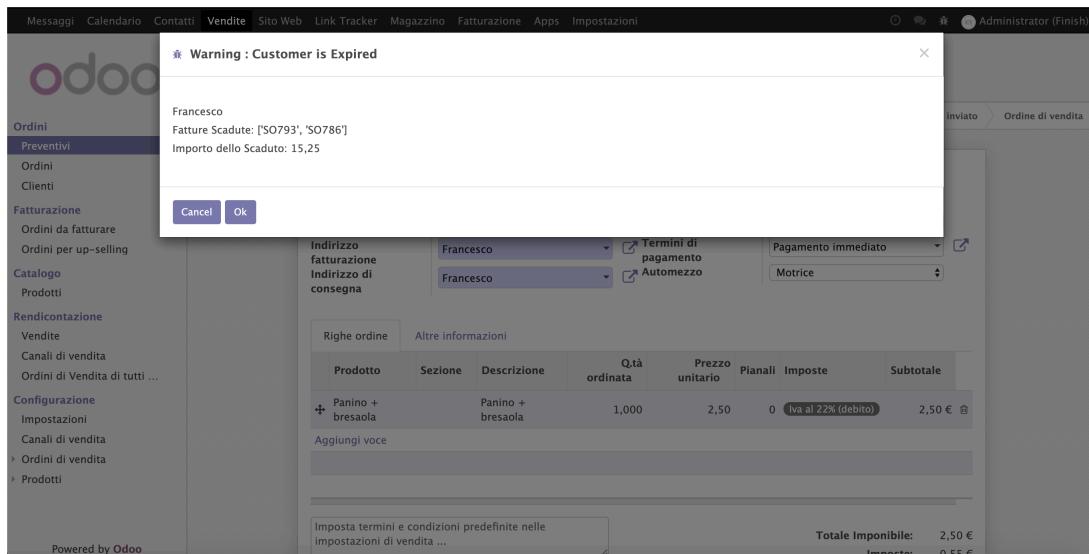


Figure 4.7: Popup blocco scaduto

4.2.2 Canali di Vendita

Nella sezione Vendite, sulla voce Canali di Vendita abbiamo la possibilità di inserire un nuovo canale. In questo modulo abbiamo aggiunto al form il campo Codice canale e Codice canale di vendita. Il campo Codice canale di vendita viene valorizzato con il codice canale + nome del canale di vendita.

Poi nella voce Clienti, che fa riferimento ad un altro modulo, quando aggiungiamo un nuovo cliente e selezioniamo *Persona*, abbiamo aggiunto il campo Codice cliente. Questo campo non verrà visualizzato nel momento in cui inserendo un nuovo nuovo cliente selezioniamo *Azienda*.

The screenshot shows the Odoo Clienti / Nuovo (New Client) form. The main title is "Clienti / Nuovo". Below it, there are two buttons: "Salva" (Save) and "Annulla" (Cancel). The form itself has several sections:

- Top section:** A radio button group between "Persona" (selected) and "Azienda". Below it are fields for "Nome" (Name) and "Codice Cliente" (Client Code), with a dropdown menu showing "Azienda". To the right are summary fields: "\$ 0 Vendite" (0 sales), a blue square icon labeled "Attivo" (Active), and "0,00 Fatturata" (0,00 invoices).
- Left side sections:**
 - "IVA": Input field with placeholder "es.: IT0533192309".
 - "Indirizzo": Input fields for "Strada/Via...", "Strada/Via 2...", "Città", "Stato", and "CAP".
 - "Tags": Input field with placeholder "tags...".
- Right side sections:**
 - "Posizione lavorativa": Input field with placeholder "ad esempio, Direttore Vendite".
 - "Telefon": Input field.
 - "Cellulare": Input field.
 - "Email": Input field.
 - "Sito Web": Input field with placeholder "per esempio www.odoo.com".
 - "Titolo": Input field.
 - "Lingua": Input field with placeholder "Italian / Italiano".
- Bottom navigation:** Tabs include "Contatti & Indirizzi" (selected), "Appunti", "Vendite & Acquisti", and "Fatturazione". A "Crea" (Create) button is located at the bottom left.

Figure 4.8: Inserimento nuovo cliente: Persona

Inserendo un nuovo cliente come *Azienda* la vista del form è differente dall'inserimento come *Persona*.

The screenshot shows the Odoo CRM interface with the following details:

- Header:** Messaggi, Calendario, Contatti, CRM, Vendite, Punto vendita, Magazzino, Fatturazione, Eventi, Dipendenti, Apps, Impostazioni, Administrator.
- Sidebar:** Ordini, Preventivi, Ordini, Clienti (selected), Fatturazione, Ordini da fatturare, Ordini per up-selling, Catalogo, Prodotti, Listini prezzi, Rendicontazione, Vendite, Canali di vendita, Ordini di Vendita di tutti ..., Configurazione, Impostazioni, Canali di vendita.
- Page Title:** Clienti / Nuovo
- Form Fields (Visible):**
 - Nome:** Input field with placeholder "Nome".
 - IVA:** Input field with placeholder "es.: IT0533192309".
 - Indirizzo:** Input fields for "Strada/Via..." and "Città", dropdowns for "Stato" and "CAP".
 - Tags:** Input field for "tags...".
 - Telefono:** Input field.
 - Cellulare:** Input field.
 - Email:** Input field.
 - Sito Web:** Input field with placeholder "per esempio www.odoo.com".
 - Lingua:** Dropdown menu with "Italian / Italiano" selected.
- Buttons:** Salva (Save), Annulla (Cancel), Crea (Create).
- Summary Bar:** Shows 0 Opportunità, 0 Meetings, \$ 0 Vendite, Attivo, 0.00 Fatturata.

Figure 4.9: Inserimento nuovo cliente: Azienda

4.2.3 Gestione imballi

Nella sezione Magazzino, sulla voce Colli di prodotto abbiamo aggiunto i seguenti campi:

- Pianale (in relazione con un solo prodotto);
- Codice Parte (in relazione con un solo prodotto);
- Bottiglie per cassa;
- N° pianali per motrice;
- N° pianali per rimorchio

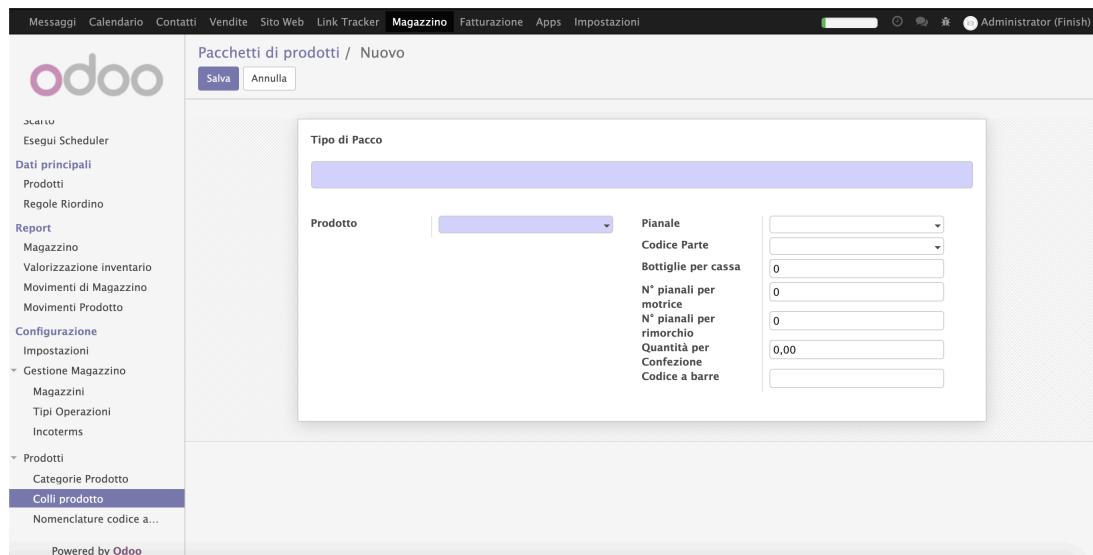


Figure 4.10: Inserimento pacchetti di prodotti

Nella sezione Vendite, voce Ordini, abbiamo aggiunto il campo *Pianali*, che viene compilato tramite la divisione tra "quantità ordinata" / "quantità per confezione". La logica per il calcolo del campo è la seguente:

- Se il rapporto da come risultato un numero intero, il sistema deve compilare il campo Pianali con il risultato del rapporto;

- Se il rapporto da come risultato un numero decimale inferiore all'unità il campo Pianali va settato a null;

(Risultato rapporto= 0,5 campo pianali = null)
- Se il rapporto da come risultato un numero decimale superiore all'unità il valore del campo Pianali va compilato con il risultato del rapporto approssimato per difetto all'intero più vicino.

(Risultato rapporto= 2,5 campo pianali = 2).

Inoltre, abbiamo inserito il seguente campo Automezzo contenente i valori: Autotreno, Motrice. Alla fine di tutto, otteniamo la seguente view.

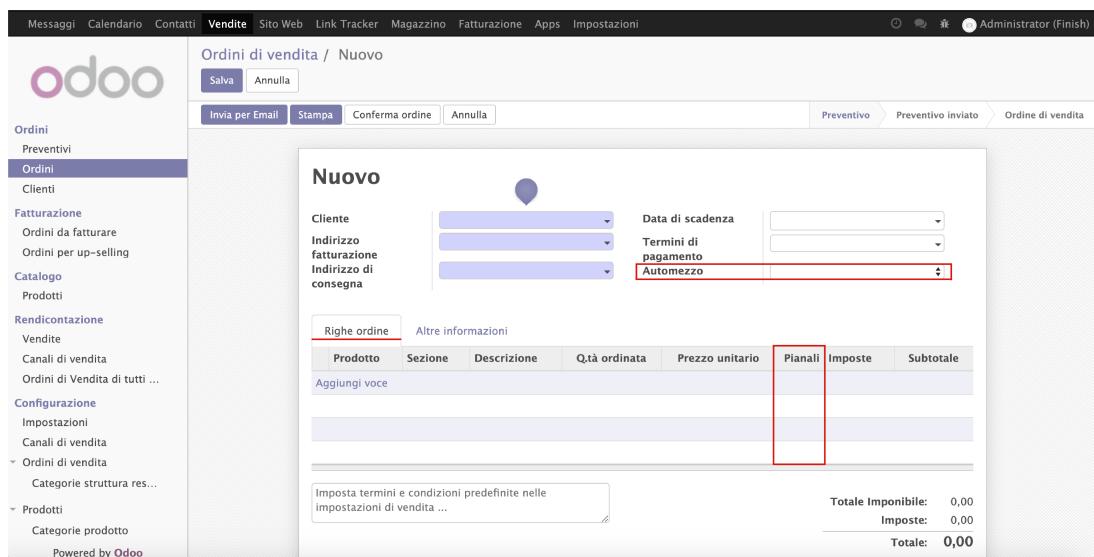


Figure 4.11: Campo Pianali e Automezzo in Ordini di vendita

Un'altra funzionalità la troviamo al salvataggio dell'ordine che innesca una funzione che effettua la seguente operazione:

- Nel caso in cui gli ordini creati nelle righe ordine presentino lo stesso prodotto occorre rendere unica la *line* aggiornando quindi la quantità;

Infine nel form di inserimento prodotto sono stati aggiunti due campi checkbox:
E' un imballo, E' un fustino.

The screenshot shows the Odoo interface for creating a new product. The top navigation bar includes links for Messaggi, Calendario, Contatti, Vendite, Sito Web, Link Tracker, Magazzino, Fatturazione, Apps, and Impostazioni. On the far right, it shows the user is 'Administrator (Finish)'.

The main window title is 'Prodotti / Nuovo'. Below the title are 'Salva' and 'Annulla' buttons. The central area is titled 'Nome prodotto' with a placeholder 'Nome prodotto'. Below this, there are two checkboxes with a red box drawn around them: 'E' un imballo' and 'E' un fustino'. There are also other checkboxes: 'Può essere venduto' (checked) and 'Può essere acquistato'.

To the right of the product name, there is a summary box with icons for Non pubblicato (Sul sito web), Movimenti Prodotto, \$ 0 Vendite, and Attivo.

Below the product name, there are tabs for Informazioni Generali, Vendite, Inventario, Fatturazione, and Note. Under 'Informazioni Generali', there are dropdowns for Tipologia prodotto (set to Consumabile), Categoria (set to All), and Riferimento interno. There are also input fields for Prezzo di vendita (1,00) and Costo (0,00). A checkbox labeled 'Costo' is checked.

The left sidebar shows a navigation tree under 'Catalogo': Ordini, Preventivi, Ordini, Clienti; Fatturazione: Ordini da fatturare, Ordini per up-selling; Catalogo: Prodotti (selected), Rendicontazione, Vendite, Canali di vendita, Ordini di Vendita di tutti ...; Configurazioni: Impostazioni, Canali di vendita, Ordini di vendita, Prodotti.

Figure 4.12: Campi checkbox in Prodotti

5

Test e validazione

Una attività di fondamentale importanza è stata quella di testing dei moduli.

5.1 Test di sistema e collaudo

Per accettare la copertura dei requisiti si è ricorso a delle prove pratiche del corretto funzionamento dell'applicazione. Questa operazione è stata svolta sia come strumento per la verifica delle operazioni durante lo sviluppo del prodotto software che come operazione di collaudo per assicurare il corretto funzionamento della piattaforma, una volta implementate tutte le funzionalità richieste.

5.1.1 Limite di credito del partner

Come detto in precedenze nel **Capitolo 4** si confronta l'importo dell'ordine di vendita da approvare con il limite di credito del partner. Se il limite di credito è inferiore oppure si modifica il prezzo unitario del prodotto, non consente di approvare l'ordine di vendita.

Per effettuare questo test, abbiamo creato un 'Ordine di Vendita' ed abbiamo compilato i seguenti campi :

- Cliente;
- Indirizzo di fatturazione;
- Indirizzo di consegna;
- Data di scadenza (successiva alla data odierna);
- Termini di pagamento;
- Automezzo (selezione fra : Autotreno e Motrice);
- Seleziono un prodotto su Righe Ordine (tra quelli esistenti o ne creo uno)

A questo punto, si fa un click sul pulsante 'Conferma ordine', e se tutto va a buon fine lo stato della fattura si sposta dallo stato 'Preventivo' allo stato 'Ordine di vendita'.

The screenshot shows the Odoo Sales module interface. The top navigation bar includes links for Messaggi, Calendario, Contatti, Vendite, Sito Web, Magazzino, Fatturazione, Apps, and Impostazioni. The current page is 'Ordini di vendita / SO837'. The main content area displays the order details for SO837. The order header includes the customer name (Francesco), delivery address (Francesco), and payment terms (Pagamento immediato, Autotreno). The order table lists one item: 'Coca cola' with a quantity of 1,000, a unit price of 5,00, and a total of 6,10 €. The status of the order is highlighted in red as 'Ordine di vendita'.

Figure 5.1: Test senza errori

Invece, se al termine della procedura di compilazione dei campi il prezzo totale dell'importo è superiore al limite di credito del partner, e faccio click su 'Conferma ordine', verrà visualizzato il popup d'errore.

In questo esempio abbiamo confermato l'ordine di 18,30 €, ma il limite di credito del partner 'Francesco' è di 10 €. Una volta che si da l'ok al popup la fattura va nello stato 'Limite di Credito', dove non è più possibile modificarla.

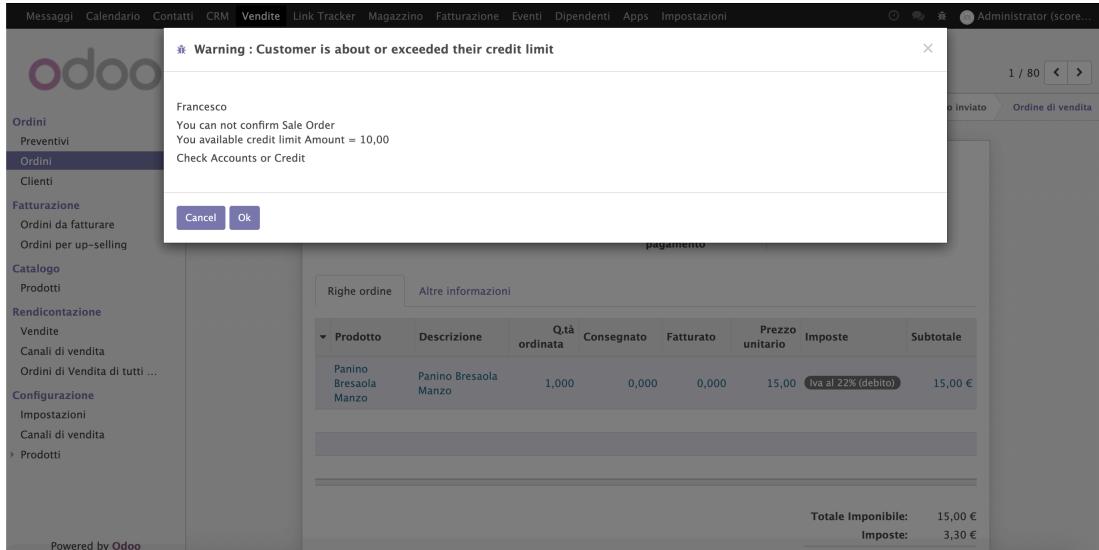


Figure 5.2: Esempio limite di credito

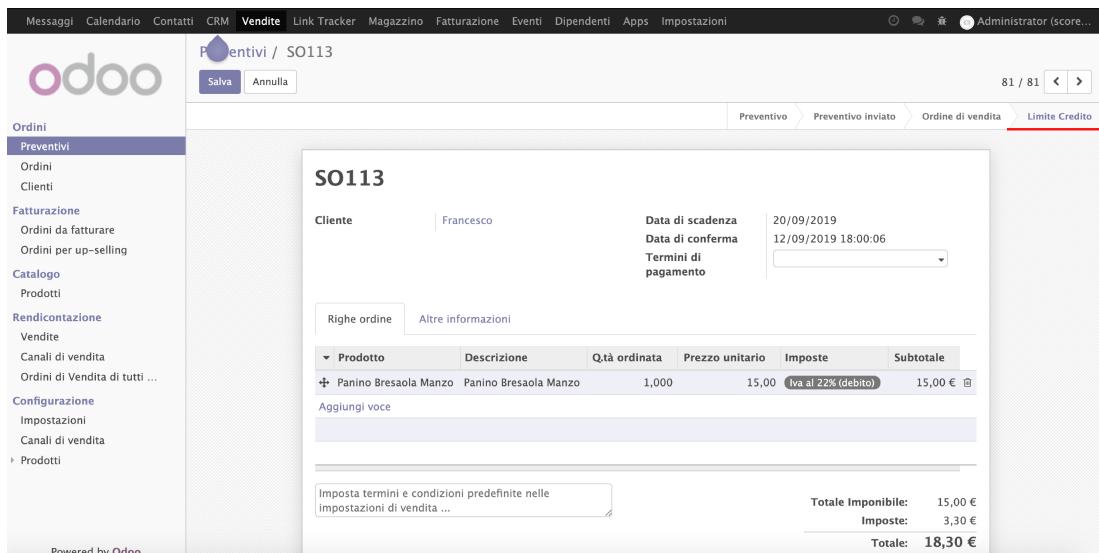


Figure 5.3: Test limite di credito

Inoltre abbiamo aggiunto un controllo al momento dell'inserimento di un prodotto. In questo esempio modifichiamo il prezzo di listino del prodotto, che è di 15,00 €, dalla vista 'Righe ordine' con 19,00 €.

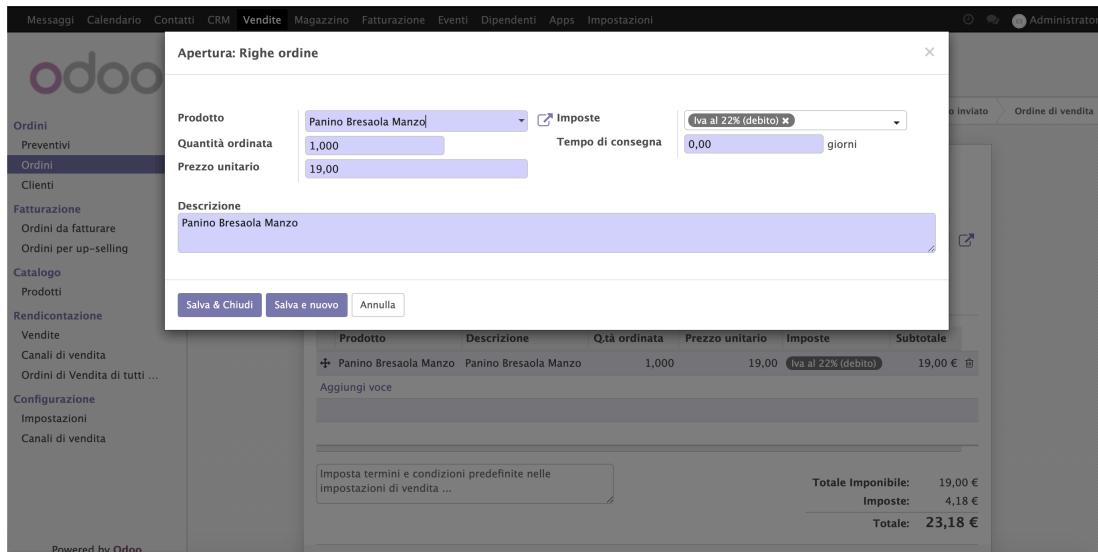


Figure 5.4: Prezzo di listino settato

Salviamo tutto, e confermiamo l'ordine. A questo punto visualizzeremo un popup d'errore che non ci permetterà di concludere l'operazione di conferma d'ordine finché non si sistemerà il prezzo.

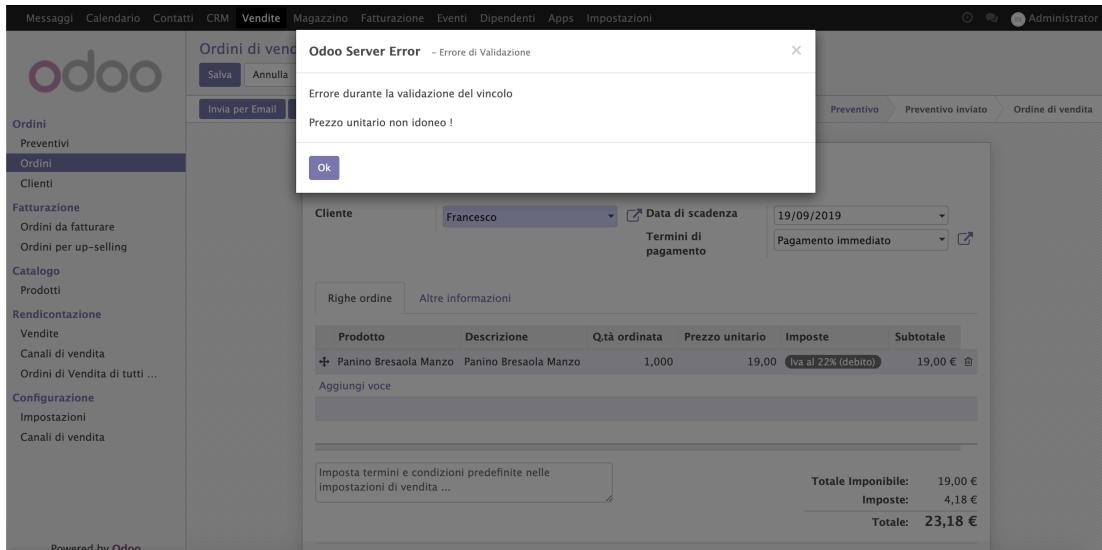


Figure 5.5: Popup d'errore sul prezzo unitario

Per quanto riguarda lo stato Blocco Scaduto, stiamo confermando un ordine con una data di scadenza antecedente a quella attuale, vuol dire che l'ordine di vendita è già scaduto.

Per effettuare questo Test, abbiamo creato un 'Ordine di Vendita' ed abbiamo compilato i seguenti campi :

- Cliente;
- Indirizzo di fatturazione;
- Indirizzo di consegna;
- Data di scadenza (antecedente alla data odierna);
- Termini di pagamento;
- Automezzo (selezione fra : Autotreno e Motrice);
- Seleziono un prodotto su Righe Ordine (tra quelli esistenti o ne creo uno)

A questo punto, si fa un click sul pulsante 'Conferma ordine', si visualizzerà il popup di errore. Una volta che si da l'ok al popup la fattura va nello stato 'Blocco Stato', dove non è più possibile modificarla.

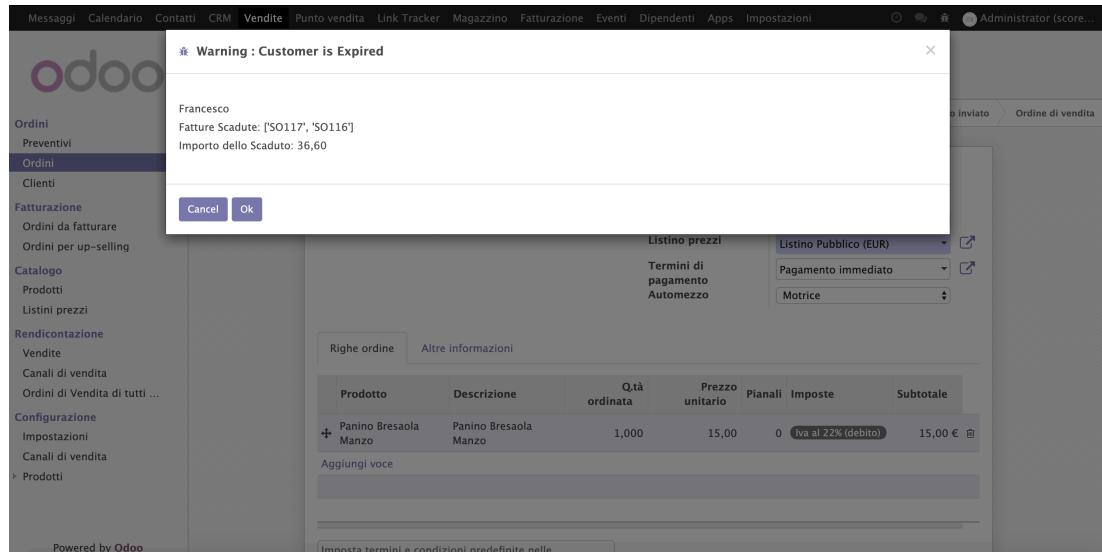


Figure 5.6: Esempio blocco scaduto

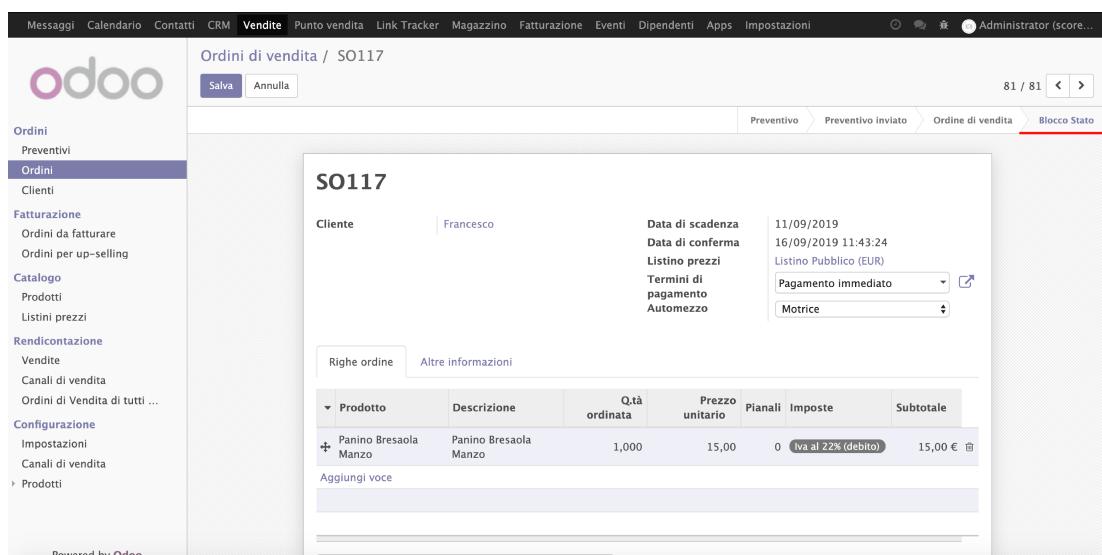


Figure 5.7: Test blocco scaduto

5.1.2 Canali di Vendita

Su questo modulo, in Vendite, nella sezione Configurazione > Canali di vendita, abbiamo verificato che il campo 'Codice canale di vendita' viene valorizzato con il campo 'Codice canale' + nome del 'Canale di vendita'.

Per effettuare questo Test, abbiamo creato un 'Canale di Vendita' ed abbiamo compilato i seguenti campi :

- Canale di Vendita;
- Codice canale;

In automatico il campo 'Codice canale di vendita' (che è *readonly*[g]), verrà valorizzato correttamente.

The screenshot shows the Odoo web interface for creating a new sales channel. The left sidebar has a tree view with categories like Ordini, Preventivi, Fatturazione, Catalogo, Rendicontazione, and Configuration. Under Configuration, 'Canali di vendita' is selected, which is also highlighted in the breadcrumb at the top: 'Canali di Vendita / Nuovo'. The main form has fields for 'Canale di Vendita' (set to 'Ferrari'), 'Codice canale' (set to '01'), 'Codice canale di vendita' (set to '01Ferrari'), 'Preventivi' (checkbox checked), 'Leader Canale' (dropdown empty), and 'Membri del Team' (button 'Aggiungi'). A 'Dashboard' button is also present. In the top right, there's a status indicator 'Attivo' with a checkmark. At the bottom left, it says 'Powered by Odoo'.

Figure 5.8: Test canale di vendita

5.1.3 Gestione imballi

Per effettuare questo test, abbiamo creato un 'Ordine di vendita' ed abbiamo compilato i seguenti campi :

- Cliente;

- Indirizzo di fatturazione;
- Indirizzo di consegna;
- Data di scadenza (successiva alla data odierna);
- Termini di pagamento;
- Automezzo (selezione fra : Autotreno e Motrice);
- Seleziono 'Aggiungi voce' su Righe Ordine

Quando abbiamo fatto click sul campo 'Aggiungi voce', che mi permette di inserire un prodotto, visualizzerò un altro form dove compilerò i seguenti campi:

- Prodotto;
- Sezione;
- Quantità ordinata (in questo esempio: 20 "numeratore");
- Confezione (selezioniamo l'esempio riportato di seguito);
- Imposte (IVA facoltativa);
- Tempo di consegna (giorni);

Come confezione abbiamo selezionato un esempio creato, di nome 'Tipo', che ha per 'Quantità per confezione'= 7 (denominatore).

Figure 5.9: Esempio form confezione

A questo punto salviamo la confezione, il prodotto, e ritorniamo all'interfaccia di 'Ordine di vendita'. Facciamo click su 'Conferma Ordine'. Come detto in precedenza, il campo 'Pianali' si compila automaticamente; è dato dal rapporto fra Quantità ordinata (20) / Quantità per confezione (7) = 2,85.

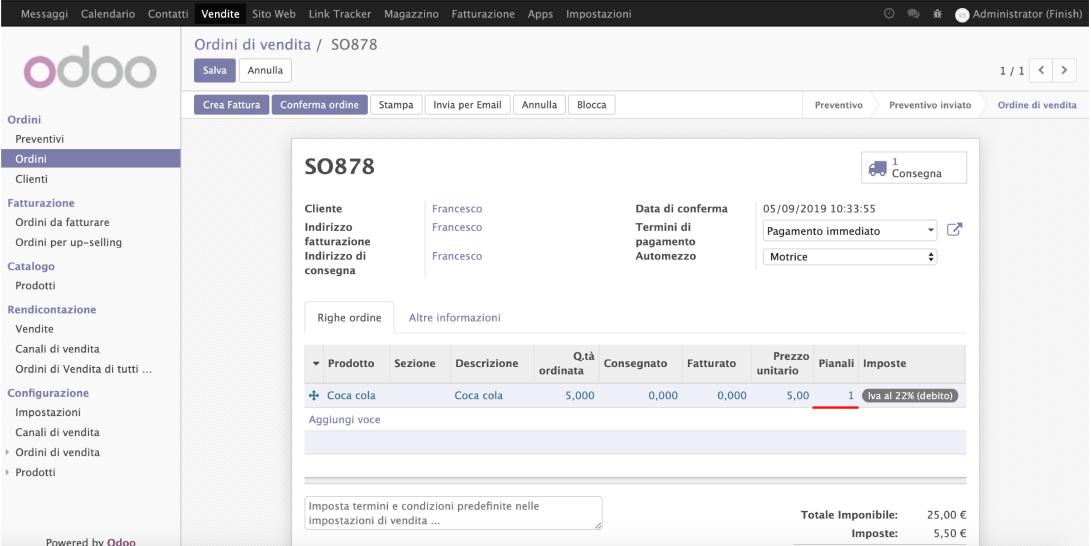
Il risultato è un numero decimale superiore all'unità, il valore del campo 'Pianali', verrà approssimato per difetto all'intero più vicino.

(Risultato rapporto = 2,85 campo pianali = 2)

The screenshot shows the Odoo Sales interface. The top navigation bar includes 'Messaggi', 'Calendario', 'Contatti', 'Vendite' (selected), 'Sito Web', 'Magazzino', 'Fatturazione', 'Apps', and 'Impostazioni'. The right side shows the user 'Administrator'. The main area displays the 'Ordini di vendita / SO850' screen. On the left, a sidebar lists categories like 'Ordini', 'Preventivi', 'Catalogo', etc. The central part shows order details for 'SO850' with a client named 'Francesco'. The 'Righe ordine' tab is selected, showing a single line item for 'Coca cola' with a quantity of 20.000, a unit price of 5.00, and a total of 100,00 €. The 'Pianali' field is highlighted in red with the value '2'. At the bottom, there are summary totals: Totale Imponibile: 100,00 €, Imposte: 22,00 €, and Totale: 122,00 €.

Figure 5.10: Primo Test campo 'Pianali'

Se il rapporto da come risultato un numero intero, il sistema deve compilare il campo Pianali con il risultato del rapporto. In questo caso abbiamo settato la Quantità ordinata (5) / Quantità per confezione (5) = 1.
 (Risultato rapporto = 1 campo pianali = 1)



The screenshot shows the Odoo Sales interface for Order SO878. The left sidebar navigation bar includes links for Messaggi, Calendario, Contatti, Vendite (selected), Sito Web, Link Tracker, Magazzino, Fatturazione, Apps, and Impostazioni. The main content area displays the order details for SO878. The order header shows the client name as 'Francesco' and the delivery address as 'Francesco'. The delivery method is set to 'Consegna' (Delivery). The payment terms are 'Pagamento immediato' (Immediate payment) and the delivery method is 'Automezzo' (Own vehicle). The order summary table lists one item: 'Coca cola' with a quantity of 5.00, a unit price of 5.00, and a total price of 25.00 €. The tax information indicates an IVA of 5.50 €. The bottom right corner of the table shows a note: 'Iva al 22% (debito)' (Tax at 22% (debit)).

Prodotto	Sezione	Descrizione	Q.tà ordinata	Consegnato	Fatturato	Prezzo unitario	Pianali	Imposte
+ Coca cola	Coca cola		5,000	0,000	0,000	5,00	1	Iva al 22% (debito)

Figure 5.11: Secondo Test campo 'Pianali'

Se il rapporto da come risultato un numero decimale inferiore all'unità il campo Pianali va settato a null. In questo caso abbiamo settato la Quantità ordinata (1) / Quantità per confezione (5) = 0,2.
 (Risultato rapporto = 0,2 campo pianali = 0)

Ordini di vendita / SO878

SO878

Prodotto	Sezione	Descrizione	Q.tà ordinata	Consegnato	Fatturato	Prezzo unitario	Pianali	Imposte
+ Coca cola		Coca cola	1,000	0,000	0,000	5,00	0	Iva al 22% (debito)

Totali: Totale Imponibile: 5,00 €
Imposte: 1,10 €

Figure 5.12: Terzo Test campo 'Pianali'

Infine al salvataggio di un ordine di vendita verifichiamo che all'inserimento di uno stesso prodotto, occorre rendere unica la *line* aggiornandone solo la quantità.

The screenshot shows the Odoo Sales Order creation interface. The left sidebar navigation bar is visible, with the 'Ordini' (Orders) item selected under 'Preventivi'. The main window displays an order with ID SO877. The order details section shows a client named 'Francesco' with delivery address 'Francesco'. The confirmation date is set to '03/09/2019 18:39:35'. The payment terms are 'Automezzo'. The order table lists two items: 'Coca cola' with a quantity of 1,000. Both lines show a unit price of 5,00, a tax of 'Iva al 22% (debito)', and a total amount of 10,00 €. The bottom right corner of the table shows a total of 12,20 €.

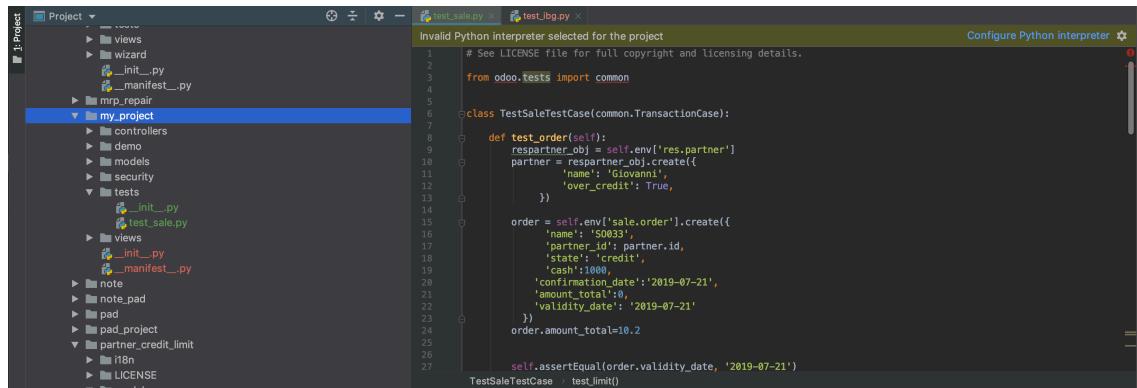
Figure 5.13: Inserimento di uno stesso prodotto

This screenshot shows the same Odoo Sales Order creation interface as Figure 5.13, but after a modification. The order table now shows a single line for 'Coca cola' with a quantity of 2,000. The unit price remains at 5,00, the tax is 'Iva al 22% (debito)', and the total amount is now 10,00 €. The bottom right corner of the table shows a total of 12,20 €. A small icon in the top right corner of the table indicates that there is one delivery note.

Figure 5.14: Test sulle *line* dei prodotti

5.2 Test di Unità

Per accettare la correttezza del codice as implemented dei modelli di ogni modulo si è ricorso ai test di *unità*^[g]. Sono stati svolti per verificare la funzione *create*, cioè quella che crea, nuovi dati da salvare nel database per ogni singolo modulo. Abbiamo testato tutti i modelli da me sviluppati.



The screenshot shows a Python code editor interface with two tabs: `test_sale.py` and `test_ibg.py`. The `test_ibg.py` tab is active and displays the following code:

```
# See LICENSE file for full copyright and licensing details.
from odoo.tests import common

class TestSaleTestCase(common.TransactionCase):
    def test_order(self):
        respartner_obj = self.env['res.partner']
        partner = respartner_obj.create({
            'name': 'Giovanni',
            'over_credit': True,
        })
        order = self.env['sale.order'].create({
            'name': 'SO103',
            'partner_id': partner.id,
            'state': 'credit',
            'cash': 1000,
            'confirmation_date': '2019-07-21',
            'amount_total': 10,
            'validity_date': '2019-07-21'
        })
        order.amount_total=10.2
        self.assertEqual(order.validity_date, '2019-07-21')
TestSaleTestCase : test_limit()
```

Figure 5.15: Test di unità

6

Conclusioni

Nel seguente capitolo verranno riportate le conclusioni che si è potuto trarre alla fine di questo stage.

6.1 Consuntivo finale

Prima di iniziare lo stage, si era concordato insieme con l’azienda la pianificazione dello stesso, definendo obiettivi da raggiungere e calendarizzandoli sulle 320 ore disponibili al fine di poterli soddisfare a pieno. La pianificazione iniziale è stata rispettata nella sua visione generale andando a modificarne invece le scadenze orarie come mostrato nella Tabella 6.1. La formazione è stata rispettata secondo il piano di lavoro, mentre nello sviluppo del primo modulo Odoo⁴ ho avuto qualche difficoltà. Questo perchè nella fase di formazione non si erano viste certe funzionalità come la creazione di un popup in caso di errore o la formazione di una template per creare l’e-mail. Questi problemi comunque sono stati pianamente risolti tramite un confronto con il tutor aziendale e i ragazzi in sede, che mi hanno fornito la giusta documentazione¹, così da permettermi di completare tutti i moduli ed effettuare i relativi test che avevamo pianificato nel tempo previsto.

Durata in ore		Descrizione dell'attività
180		A: Formazione
	8	<ul style="list-style-type: none"> • Studio Database Sql-PostgreSQL
	24	<ul style="list-style-type: none"> • Studio XML-XPath
	4	<ul style="list-style-type: none"> • Studio Html5/CSS3
	16	<ul style="list-style-type: none"> • Studio Javascript
	40	<ul style="list-style-type: none"> • Studio linguaggio Python
	32	<ul style="list-style-type: none"> • Studio Piattaforma Erp Odoo
	56	<ul style="list-style-type: none"> • Sviluppo primo modulo di prova
100		B: Sviluppo moduli
	100	<ul style="list-style-type: none"> • Implementazione di nuovi moduli Odoo in ambito Food-and-beverage
40		C: Collaudo Finale
	28	<ul style="list-style-type: none"> • Test
	4	<ul style="list-style-type: none"> • Stesura Documentazione Finale
	4	<ul style="list-style-type: none"> • Collaudo e consegna del codice
Totale: 320		

Table 6.1: Ripartizione reale delle ore di stage

6.2 Raggiungimento degli obiettivi

Gli obiettivi concordati prima dell'inizio dello stage e riportati nel **Capitolo 2**, prevedevano il soddisfacimento di quattro obiettivi obbligatori, due desiderabili e uno opzionale. Gli obiettivi concordati sono stati soddisfatti a pieno tranne l'obiettivo opzionale dove, visto il poco tempo rimasto si è deciso di eseguire i test di unità nei modelli dei moduli sviluppati, così da verificare al meglio il codice.

6.3 Conoscenze acquisite

Le conoscenze acquisite durante il corso dello stage sono state le seguenti:

- Linguaggio Python¹¹;
- pgAdmin 4⁷;
- Librerie ORM Api⁶;
- Costruzione modulo Odoo⁴;
- Package management attraverso Homebrew²;
- Package management attraverso PIP⁸;
- Approfondimento linguaggio XML/XPATH

6.4 Sviluppi futuri del software

Odoo è in continuo e rapido sviluppo. Esistono centinaia di moduli sviluppati dalla comunità e dalla casa madre liberamente utilizzabili, quindi a volte le cose che non riusciamo a trovare forse sono disponibili facendo le ricerche giuste.

I clienti per lavorare con Odoo non necessitano di installare alcun software sul proprio pc ma usano un normale programma per la navigazione internet su un computer collegato al server, accedendo tramite inserimento di nome utente e password. È un software basato soprattutto sulla fatturazione, la gestione dell'inventario ma come ho visto in azienda, è molto richiesto nella gestione di e-commerce. Lo sviluppo futuro sarà sicuramente quello di creare informazioni chiare e che trasmettono fiducia al cliente in fase di acquisto.

6.5 Valutazione personale

La valutazione dello stage svolto presso Sync Lab non può che dirsi positiva, ho trovato un ambiente accogliente e dei colleghi sempre disponibili. Tecnicamente parlando affrontare un problema così complesso è stato formativo, poiché ho imparato ad usare strumenti utili ed un linguaggio molto popolare ed usato al giorno d'oggi.

I lunghi confronti con i colleghi sono stati importanti per capire l'approccio migliore per affrontare i vari task.

Lo stage è stato la conclusione di un percorso di studio molto intenso e credo si possa dire abbia messo un punto a tutti gli sforzi fatti e a mille pensieri per raggiungere l'obbiettivo.

Glossario

- **Brainstorming:** incontri di gruppo creativi utili per risolvere problemi o individuare nuove idee. Servono più di due partecipanti in modo che vi sia una discussione arbitraria e quindi utile.
- **Ciclo Di Vita:** insieme degli stati che il prodotto assume dal concepimento al ritiro.
- **Food-and-beverage:** settore riguardante la ristorazione, alimentare cioè cibi e bevande
- **Incrementale:** ciascun modulo eredita struttura dal modello e viste dai precedenti, quindi si procede per aggiunte ad un impianto base
- **Modello Evolutivo:** modello di ciclo di vita che aiuta a rispondere a bisogni non inizialmente preventivabili, può richiedere il rilascio di più versioni esterne attive in parallelo. La grande differenza con il modello incrementale è che l'analisi e progettazione iniziali vengono ripetute.
- **Milestone:** punto nel tempo al quale associamo un insieme di stati di avanzamento.
- **Open Source:** termine utilizzato per riferirsi ad un software di cui i detentori dei diritti sullo stesso ne rendono pubblico il codice sorgente.
- **Project Management:** gestione delle attività di analisi, progettazione, pianificazione e realizzazione degli obiettivi di un progetto, compito svolto dal project manager di un'azienda attraverso anche strumenti idonei.
- **Readonly:** è un campo di sola lettura, dove non è possibile scrivere dati
- **RPC:** è un protocollo che un programma può utilizzare per richiedere un servizio da un programma situato in un altro computer di una rete, senza dover comprendere i dettagli della rete. Utilizza il modello client-server. Il programma richiedente è un client e il servizio che fornisce il programma è il server.
- **Slack:** tempo aggiuntivo ad un'attività che ha lo scopo di evitare ritardi nella produzione del prodotto.

- **Time-to-Market:** tempo che intercorre fra l'inizio del processo di sviluppo di un nuovo prodotto e l'avvio della sua commercializzazione.
- **Unità:** Per unità si intende normalmente il minimo componente di un programma dotato di funzionamento autonomo e producibile da un singolo programmatore

Lista degli acronimi

ORM Object-Relational Mapping

RPC Remote Procedure Call

Riferimenti bibliografici

- [1] documentazione. URL <https://www.odoo.com/documentation/11.0/index.html>.
- [2] Homebrew. URL https://brew.sh/index_it.
- [3] Javascript. URL <https://www.javascript.com/>.
- [4] Odoo. URL https://www.odoo.com/it_IT/.
- [5] Openproject. URL <https://www.openproject.org/>.
- [6] Orm api. URL <https://www.odoo.com/documentation/11.0/reference/orm.html>.
- [7] pgadmin 4. URL <https://www.pgadmin.org/>.
- [8] Pip. URL <https://pypi.org/project/pip/>.
- [9] Postgresql. URL <https://www.postgresql.org/>.
- [10] Pycharm. URL <https://www.jetbrains.com/pycharm/>.
- [11] Python. URL <https://www.python.org/>.
- [12] Slack. URL <https://slack.com>.
- [13] Stageit. URL <http://informatica.math.unipd.it/laurea/stageit.html>.