

POLITECNICO DI TORINO

Corso di Laurea Magistrale in

Ingegneria Gestionale

Tesi di Laurea Magistrale



**Politecnico  
di Torino**

Applicazioni low code e sviluppo di un ambiente  
per la geolocalizzazione della flotta auto aziendale  
e la gestione del relativo workflow approvativo

**Relatore:**

Prof. Luca Ardito

**Co-Relatore:**

Morassutto Simone

**Candidato:**

Lorenzo Riitano

Anno accademico 2022/2023

## Sommario

Il presente elaborato intende offrire un'accurata esposizione delle attività di analisi e realizzazione di una web application per la prenotazione e la geolocalizzazione della flotta auto aziendale e per la gestione del relativo workflow approvativo sviluppata autonomamente dal tesista durante un'un'esperienza di tirocinio presso l'azienda Mediamente Consulting S.r.l.

Partendo da un'analisi approfondita della situazione in essere e degli stakeholders coinvolti sono stati definiti ed implementati il modello dei dati e le logiche di funzionamento dell'applicazione con particolare focus sugli step del processo di approvazione.

E' stata analizzata e sviluppata una funzionalità di geolocalizzazione per memorizzare all'interno del database le coordinate geografiche delle auto aziendali e per consentire il reindirizzamento degli utenti ad una pagina di Google Maps che segnali il percorso per raggiungere un'auto. Come funzionalità a supporto della gestione del parco auto aziendale è stato implementato un cruscotto per fornire informazioni di servizio (scadenza assicurazione, revisione, tassa automobilistica ecc.) ed una chat per consentire agli utilizzatori di segnalare eventuali problematiche incorse durante l'utilizzo di ciascuna vettura.

L'applicazione è stata realizzata con lo strumento Oracle Application Express (APEX), una delle più diffuse piattaforme low code per lo sviluppo di applicazioni web in ambito business enterprise.

Mediante l'utilizzo della piattaforma Oracle Apex è stato possibile approfondire le conoscenze sul linguaggio di interrogazione SQL (Structured Query Language), frequentemente impiegato per l'interazione con Database Management Systems (DBMS), sul linguaggio di programmazione procedurale PL/SQL funzionale anch'esso all'interrogazione di database e sui linguaggi HTML, CSS e JavaScript.

Infatti, sebbene una piattaforma di sviluppo low code consenta di realizzare applicazioni in modo più semplice e veloce rispetto all'utilizzo del codice tradizionale, la dimestichezza con i linguaggi appena elencati rappresenta un requisito indispensabile per un APEX developer.

L'elaborato offre dapprima un'analisi delle principali caratteristiche della metodologia di sviluppo Low-Code concentrando l'attenzione sui benefici ma anche sulle carenze rispetto alle tecniche di sviluppo tradizionale basate interamente sulla scrittura di codice. Successivamente viene fornita una panoramica sul mercato attuale delle piattaforme Low-Code con un focus particolare sulle caratteristiche dei principali player e sui relativi punti di forza e di debolezza.

L'analisi delle soluzioni presenti sul mercato si conclude con una disamina dei requisiti significativi per una piattaforma Low Code di classe enterprise che deve possedere funzionalità e capacità di gestione della complessità decisamente superiori rispetto alle piattaforme destinate ad un pubblico consumer offrendo soluzioni anche per applicazioni mission critical.

Poiché il progetto realizzato durante il tirocinio aziendale è stato sviluppato in ambiente Oracle APEX, un intero capitolo del presente elaborato è dedicato alla descrizione della suddetta piattaforma ponendo l'accento sulla sua natura di tool per uno sviluppo dichiarativo, web-based e data-centric. In questa sezione viene analizzata l'architettura a tre livelli di Oracle APEX, viene descritta l'anatomia di una applicazione (che risiede in una precisa area di lavoro denominata "workspace"), vengono esaminati i componenti dell'ambiente di sviluppo di tipo Integrated Development Environment (IDE) messo a disposizione da Oracle APEX e vengono descritte le principali funzionalità offerte dalla piattaforma.

Si passa successivamente alla descrizione dettagliata del progetto realizzato partendo dall'analisi dei requisiti espressi dall'azienda Mediamente Consulting S.r.l. In questo capitolo vengono illustrate le soluzioni implementate per ciascun requisito

con approfondimenti sulle funzionalità di APEX utilizzate nelle varie fasi di sviluppo.

Per finire vengono proposti alcuni possibili sviluppi futuri dell'applicazione e vengono espresse delle osservazioni conclusive sulla personale esperienza di sviluppo mediante la piattaforma low code utilizzata.

# Indice

<b>Elenco delle figure</b>	6
<b>1 Sviluppo low code</b>	9
1.1 Sviluppo low code: definizione e caratteristiche . . . . .	9
1.2 Traditional vs low code development . . . . .	15
1.3 Approccio Data driven . . . . .	22
1.3.1 Complementarietà dell’approccio data-driven e dello sviluppo low-code . . . . .	25
<b>2 Panoramica sul mercato delle piattaforme di sviluppo low code</b>	29
2.1 Mercato delle piattaforme di sviluppo Low Code. . . . .	29
2.2 Attori del mercato . . . . .	32
2.3 Sfide che il settore dovrà affrontare . . . . .	39
<b>3 Oracle APEX</b>	43
3.1 Architettura Oracle Apex . . . . .	45
3.2 Oracle Application Express Development Environment . . . . .	47
3.2.1 Application builder . . . . .	49
3.2.2 SQL Workshop . . . . .	50
3.2.3 Team development . . . . .	54
3.2.4 Gallery . . . . .	54
3.3 Le applicazioni in Oracle Apex . . . . .	55

3.3.1	Page Designer . . . . .	57
<b>4</b>	<b>MMPrenotazioni Auto</b>	<b>61</b>
4.0.1	Il contesto. . . . .	61
4.0.2	Analisi dei requisiti. . . . .	62
4.0.3	Stakeholders . . . . .	63
4.0.4	Progettazione base dati . . . . .	63
4.0.5	Modello di funzionamento dell'applicazione . . . . .	69
4.1	Progettazione del sistema di prenotazione e relativo flusso approvativo.	75
4.1.1	Pagina nuova prenotazione . . . . .	75
4.1.2	Definizione del Task di avvio workflow approvativo . . . . .	79
4.1.3	Unified task list . . . . .	88
4.1.4	Ciclo di vita di un Task . . . . .	91
4.1.5	Progettazione Sistema di aggiornamento delle coordinate di un'auto . . . . .	94
4.1.6	Procedura di accesso alle coordinate dell'auto . . . . .	99
4.2	Sviluppi futuri . . . . .	105
<b>5</b>	<b>Conclusioni</b>	<b>107</b>
	<b>Bibliografia</b>	<b>109</b>

# Elenco delle figure

1.1	Confronto traditional vs low code approach. . . . .	17
1.2	Percezione degli sviluppatori riguardo la rapidità di sviluppo della metodologia low code confrontata con la metodologia tradizionale. [1]	19
1.3	Utilizzo della metodologia low code per ambito. [2] . . . . .	21
1.4	Utilizzo delle piattaforme Low Code/No Code per settore [3] . . . .	22
1.5	Caratteristiche approccio data-driven. . . . .	25
1.6	Complementarietà degli approcci. . . . .	26
2.1	Revenues delle piattaforme di sviluppo low code [4] . . . . .	30
2.2	Qualità percepita dagli sviluppatori nel 2021. [5] . . . . .	31
2.3	Gartner low code platforms. [6] . . . . .	33
2.4	Opportunità di mercato lowcode entro il 2027 secondo Gartner [7] .	41
3.1	Architettura di un ambiente con Oracle Apex. [8] . . . . .	44
3.2	Architettura Oracle Apex. [9] . . . . .	46
3.3	Workspace Oracle Apex . . . . .	47
3.4	Workspace Oracle Apex [8] . . . . .	48
3.5	App Builder Oracle Apex. . . . .	50
3.6	SQL workshop. . . . .	51
3.7	Object browser Oracle Apex. . . . .	52
3.8	SQL commands. . . . .	52

3.9	Esempio QuickSQL. [10]	53
3.10	Anatomia applicazione Oracle Apex. [11]	56
3.11	Page designer.	57
4.1	Foglio excel prenotazioni auto.	62
4.2	Base dati prenotazioni auto.	64
4.3	Branches dell'applicazione.	70
4.4	Vista pagina di calendario.	71
4.5	Mappa per raggiungere l'auto desiderata.	72
4.6	Pagina per l'aggiornamento coordinate.	73
4.7	BPMN flusso approvativo.	75
4.8	Validations nuova prenotazione	76
4.9	Definizione task.	82
4.10	Definizione task.	82
4.11	Azione eseguita all'approvazione del task.	85
4.12	Processo di creazione task.	86
4.13	Dettagli processo creazione task.	87
4.14	Pagina Le mie richieste.	89
4.15	Pagina approvazioni da effettuare.	89
4.16	Pagina Task details.	90
4.17	Ciclo vita di un task in Oracle Apex [12]	93
4.18	Attori ed azioni di un task in Oracle Apex. [13]	94
4.19	BPMN aggiornamento coordinate.	95
4.20	Pagina aggiornamento coordinate.	95
4.21	BPMN accesso auto.	99
4.22	Interactive grid pagina Auto.	100
4.23	Cards per accedere alla chat relativa all'auto a cui si riferisce la card.	101
4.24	Regione per consultare/rilasciare commenti.	102
4.25	Regione per ottenere le coordinate dell'auto.	103



# Capitolo 1

## Sviluppo low code

### 1.1 Sviluppo low code: definizione e caratteristiche

Il Low Code Development è una metodologia che permettere di semplificare notevolmente il processo di sviluppo software offrendo la possibilità di operare in tale ambito anche a persone prive di particolari competenze di programmazione (sviluppatori “non professionisti”) e fornendo contemporaneamente agli sviluppatori più esperti (i cosiddetti sviluppatori professionisti) una soluzione di sviluppo più rapida e meno impegnativa. Le piattaforme low code rappresentano uno strumento per creare applicazioni aziendali riducendo al minimo la programmazione tradizionale cioè la scrittura di codice manuale.

Questo tipo di sviluppo si presenta dunque come metodologia in grado di "democratizzare" la scrittura di applicazioni informatiche rendendola una pratica fruibile da parte di una platea di popolazione sempre più ampia. [7] Tale metodologia si basa sull'utilizzo di piattaforme di sviluppo low code (Low-Code Development Platform – LCDP) vale a dire su ambienti che utilizzano tecniche di progettazione basata su modelli e su strumenti di programmazione visiva e di generazione automatica del codice. [14] Le tecniche di implementazione Low Code prevedono infatti l'utilizzo di

sistemi di rappresentazione visuale e di configuratori per esprimere il funzionamento che si desidera ottenere lasciando alle piattaforme di sviluppo la realizzazione vera e propria delle applicazioni.

Diventa quindi possibile creare del software applicativo utilizzando moduli di configurazione e interfacce grafiche predefiniti all'interno della piattaforma, anziché fare ricorso alle tecniche di programmazione tradizionale basate sulla scrittura di codice sorgente.

Le suddette piattaforme offrono in maniera integrata molteplici funzioni tra le quali:

- **Strumenti visivi e interfacce grafiche:** che consentono agli sviluppatori la definizione delle interfacce utente e l'impostazione delle logiche delle applicazioni mediante azioni di drag and drop di elementi già precostituiti e l'utilizzo di strumenti visivi.
- **Integrazione dei dati:** le piattaforme di sviluppo low code consentono alle applicazioni di accedere ed utilizzare dati provenienti da fonti differenti come database, API o servizi web.
- **Scalabilità:** nonostante la creazione di applicazioni sia semplificata, tali applicazioni possono essere in qualsiasi momento integrate con ulteriori pagine o funzioni per aumentarne i casi di utilizzo.
- **Componenti predefiniti:** le piattaforme di sviluppo low code mettono a disposizione degli sviluppatori una serie di librerie di componenti predefiniti come pulsanti, layout di forms, calendari e molti altri items utili in fase di sviluppo.

Lo sviluppo tradizionale delle applicazioni richiede l'utilizzo di professionisti altamente specializzati nella scrittura di codice informatico e spesso la necessità di personale esperto comporta una saturazione delle risorse del reparto IT con conseguente dilatazione dei tempi di consegna. Al proposito, secondo un rapporto di

International Data Corporation, già nel 2021 mancavano 1.4 milioni di sviluppatori per soddisfare le esigenze del mercato. Le piattaforme Low Code sono invece progettate appositamente per gli utenti che non hanno esperienza di codifica ma che hanno dimestichezza con i processi e i flussi di lavoro (i cosiddetti workflow) all'interno del proprio reparto aziendale; questa caratteristica consente di sfruttare le conoscenze di processo dei dipendenti delle singole unità produttive e di realizzare in breve tempo applicazioni in grado di meglio rispondere alle esigenze dei vari reparti.

Un ulteriore beneficio che le aziende possono trarre dall'utilizzo di una metodologia di sviluppo low code è rappresentato dal maggior grado di collaborazione tra lo sviluppatore software e gli altri dipendenti aziendali; questi ultimi, pur non avendo un background completo in materia di sviluppo software, potranno infatti partecipare attivamente al processo di implementazione e personalizzazione delle applicazioni con l'obiettivo di ottenere funzionalità che meglio si adattano alle esigenze operative e di processo del loro reparto. Questo tipo di approccio si fa particolarmente evidente quando si adotta una metodologia "Agile" per la realizzazione di un progetto.

Alcuni dei passaggi fondamentali di una metodologia di sviluppo Low Code possono essere brevemente sintetizzati come segue: [14]

- **Definizione delle esigenze e del risultato desiderato:** proprio perché strumento di sviluppo destinato anche ad utenti non esperti in tecniche di programmazione, nella fase iniziale del progetto è utile porsi alcune domande volte ad identificare le esigenze aziendali e i risultati desiderati. Occorrerà quindi individuare il problema che l'applicazione dovrà risolvere, quali saranno gli utilizzatori, di quali informazioni e dati avrà bisogno per funzionare ecc.

- **Elaborazione di un processo di business o workflow:** tramite gli strumenti messi a disposizione dalle piattaforme Low Code, gli utenti specificano e documentano i processi e i workflow desiderati. Occorre pertanto individuare i moduli necessari all'interno dell'applicazione (in funzione dello scopo che devono garantire) che sarà opportuno sviluppare come entità indipendenti. Ad esempio, alcuni moduli possono raccogliere dati, altri possono innescare un'azione o un evento ecc.
- **Test e implementazione del progetto sotto forma di applicazione Low Code:** spesso la piattaforma Low Code è in grado di risolvere tutte le complessità back-end cioè di elaborazione vera e propria dei dati.

Le piattaforme Low-Code offrono in genere una gamma abbastanza ampia di casi d'uso in vari settori; alcuni scenari comuni in cui vengono impiegate sono i seguenti: [15]

- **Automazione dei processi aziendali:** vengono utilizzate sempre più spesso per automatizzare attività ripetitive e banali, per snellire i processi aumentando l'efficienza complessiva.
- **Sistemi di gestione delle relazioni con i clienti (CRM):** la personalizzazione dei sistemi CRM è un altro caso di utilizzo frequente delle piattaforme low-code che consentono alle aziende di adattare il CRM (Customer Relationship Management) alle loro esigenze specifiche, migliorando il servizio e le relazioni con i clienti.
- **Sviluppo di applicazioni mobili:** generalmente sono compatibili con la visualizzazione su dispositivi mobile; le interfacce drag-and-drop e i modelli precostituiti facilitano uno sviluppo rapido di applicazioni destinate a questo tipo di device.

- **Prototipazione rapida:** sono particolarmente utilizzate per creare prototipi al fine di testare rapidamente nuove idee e rispondere in tempi brevi ai cambiamenti del mercato.
- **Raccolta e analisi dei dati:** possono essere utilizzate per creare applicazioni che raccolgono e analizzano i dati.
- **Modernizzazione dei sistemi legacy:** consentono di creare interfacce moderne verso i sistemi legacy di un'azienda evitando interventi onerosi per la loro sostituzione e prolungandone di fatto la vita utile.

In relazione al sempre più diffuso ricorso delle aziende a metodologie di sviluppo software basato su piattaforme Low Code, secondo due recenti analisi di Gartner e Forrester, il mercato dello sviluppo informatico sta attraversando una vera e propria rivoluzione e si ritiene che nei prossimi anni le applicazioni low code rappresenteranno la modalità di sviluppo prevalente nelle aziende. In particolare, secondo Gartner, entro il 2024 le applicazioni verranno sviluppate per il 65% mediante strumenti low code. A rafforzamento di questa tesi, in un articolo pubblicato su Forbes nel maggio del 2021 Asokan Ashok (CEO di UnfoldLabs), afferma che *“la combinazione di innovazione low-code e convenzionale sarà adottata dal 75% delle aziende. Entro il 2030, si prevede che il mercato globale delle piattaforme di sviluppo low-code genererà entrate per 187 miliardi di dollari.”* [16]

Un elemento che spesso risulta in difetto nelle metodologie di sviluppo Low Code rispetto alla metodologia di sviluppo tradizionale è rappresentato dall'aspetto di personalizzazione che risulta soddisfatto pienamente per quanto riguarda lo sviluppo tradizionale mentre talvolta viene sacrificato dalle piattaforme di sviluppo low code a favore della semplicità. Sempre secondo Asokan Ashok, lo sviluppo Low Code non richiede di dotarsi di personale altamente specializzato ma, nel caso in cui sia abbia bisogno di una funzionalità specifica non disponibile sulla piattaforma potrebbe essere necessario sviluppare del codice personalizzato e talvolta unire il

codice personalizzato a quello low-code può risultare molto più costoso che creare una soluzione completamente personalizzata fin dall'inizio.

## 1.2 Traditional vs low code development

Nella seguente sezione verranno analizzate le differenze tra l'approccio di sviluppo tradizionale e la metodologia low code.

Quando si parla di processo di sviluppo software è necessario considerare alcune metriche fondamentali che ne determinano l'efficienza e l'efficacia. Per quanto riguarda la produzione del software, come per ogni processo produttivo, una particolare attenzione viene posta al cosiddetto Time to Market, definito come periodo che intercorre tra l'inizio dello sviluppo e l'uscita del prodotto o servizio sul mercato.

Il Time to Market risulta molto importante per una azienda in quanto può determinare un vantaggio competitivo in relazione ai competitors. In relazione allo sviluppo software il Time to Market risulta strettamente connesso alle linee di codice scritte: un maggior numero di linee di codice corrisponderà ad un numero maggiore di ore da impiegare per la scrittura e il debug del codice stesso. Tutto ciò potrebbe portare ad un Time to Market significativamente più elevato oltre che rappresentare ovviamente un maggior costo per l'azienda.

Un ulteriore aspetto di primaria importanza nel processo di sviluppo software è rappresentato dalla cosiddetta "riusabilità del codice" cioè dalla realizzazione di parti di codice che possono essere richiamate in altri programmi senza la necessità di riscriverle ogni volta daccapo. A tale proposito, realizzando un software applicativo dovrebbe essere prassi abituale scomporre il codice in moduli ai quali viene associata una determinata funzione; più il modulo è generico più sarà possibile per lo sviluppatore adattare il modulo scritto in precedenza ad un progetto differente, risparmiando così la fatica ed il tempo dedicato allo sviluppo ed al debug del codice. [17]

I vari moduli sviluppati vengono generalmente raggruppati in librerie che possono essere a loro volta "collegate" all'interno dei vari programmi. Questa metodologia

è particolarmente utilizzata nella pratica della programmazione ad oggetti, nella quale le “classi” possono essere "richiamate", "importate" o "ereditate" all'interno di un programma in base alle necessità. E' intuitivo comprendere che il riuso di codice consente di ottenere un notevole risparmio nei tempi di scrittura di un programma da parte del programmatore che può richiamare componenti e/o funzionalità già sviluppate in precedenza migliorando nel contempo la leggibilità del codice che diventa più corto e sintetico.

Il fatto di disporre di codice pronto all'uso, e di intervenire nella scrittura soltanto in minima parte nelle modifiche e nelle integrazioni richieste nello specifico da ciascun progetto è un approccio molto utilizzato nelle applicazioni moderne che sono spesso assemblate grazie a componenti tra loro disaccoppiati, sul modello dell'architettura a microservizi. [18]

Un aspetto quasi contrastante con quello appena descritto è rappresentato dalla personalizzazione del codice quindi dalla “flessibilità” dello strumento; assumendo infatti che lo sviluppatore abbia compreso a pieno i requisiti del cliente, maggiore libertà in termini di personalizzazione viene data allo sviluppatore e maggiore sarà la probabilità che il prodotto incontri le esigenze del cliente finale. Come già detto, le piattaforme Low Code offrono scarsi margini di personalizzazione e questo può rappresentare un fattore negativo per quelle applicazioni aziendali che richiedono task e caratteristiche particolari e il cui sviluppo, generalmente di natura più complessa, dovrebbe continuare ad essere demandato a programmatori professionisti in grado di salvaguardare anche le caratteristiche di performance e di sicurezza del prodotto finale.

	TRADITIONAL	LOW CODE
COMPETENZE RICHIESTE	✗	✓
TTM	✗	✓
COSTI	✗	✓
RIUSABILITA' DEL CODICE	✓	✓
FLESSIBILITA'	✓	—

Figura 1.1. Confronto traditional vs low code approach.

La figura sopra riportata mostra il confronto tra i due approcci prendendo in considerazione le metriche appena descritte. Già da una prima analisi è possibile notare che i due approcci hanno caratteristiche differenti e, di conseguenza, l'opportunità di utilizzo di ciascuna delle due metodologie dovrà essere valutata caso per caso in base alle differenti situazioni da affrontare.

In linea teorica i due approcci non si escludono, se mai dovrebbero completarsi a vicenda, sfruttando i rispettivi vantaggi operativi e procedurali.

Ad oggi, i casi più comuni di utilizzo dell'approccio low code includono: [19]

- *Portali e applicazioni per clienti e partner*
- *Applicazioni internet*
- *Business process automation*
- *Modernizzazione di applicazioni obsolete*

Sul tema del Time To Market è interessante osservare, come riportato in figura 1.2, che ben il 74% degli sviluppatori intervistati ritiene che l'approccio Low Code comporti un risparmio di tempo tra il 20% e l'80% con un picco intorno il 50% il che significa che la popolazione intervistata percepisce di poter ottenere lo stesso risultato nella metà del tempo rispetto all'approccio di sviluppo tradizionale. Inoltre, il processo di sviluppo e distribuzione risulta semplificato e consente di rendere operative le nuove soluzioni in pochi giorni anziché in settimane o mesi.

Quando si parla di costo dello sviluppo secondo l'approccio low code un elemento da non sottovalutare è quello dei costi di licenza che le aziende devono sostenere per poter utilizzare le piattaforme low code in quanto, nella quasi totalità dei casi il loro utilizzo è concesso in cambio di un corrispettivo in denaro. Negli ultimi anni le aziende hanno comunque riscontrato che generalmente il costo sostenuto per la licenza è ampiamente compensato dal risparmio ottenuto in termini di ore uomo necessarie per la realizzazione delle varie attività.

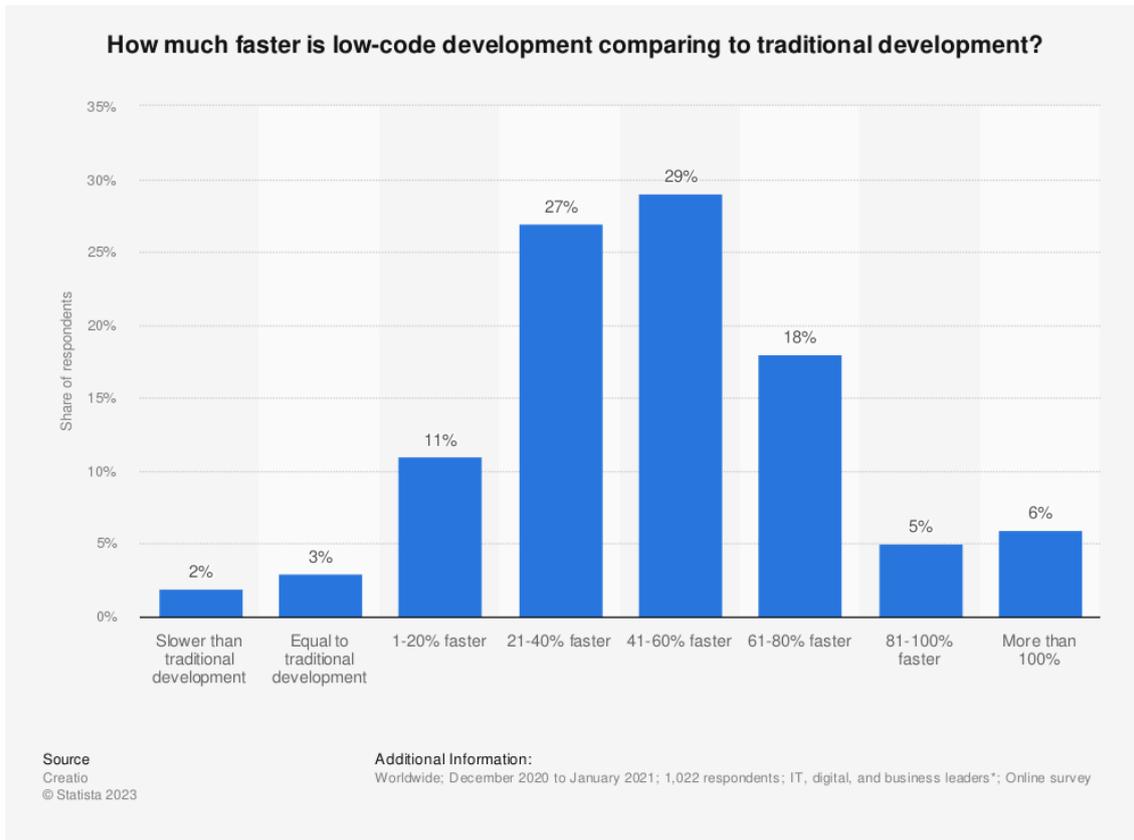


Figura 1.2. Percezione degli sviluppatori riguardo la rapidità di sviluppo della metodologia low code confrontata con la metodologia tradizionale. [1]

Un altro aspetto da tenere in considerazione quando si valuta l’approccio Low Code rispetto a quello tradizionale è l’ostacolo costituito dal cosiddetto “vendor lock-in” vale a dire la situazione di dipendenza che si verifica quando un’azienda è vincolata al fornitore di una piattaforma closed-source al punto tale da non poterla più sostituire o modificare qualcosa al suo interno senza gravi conseguenze. [20]

Un interessante elemento a favore dell’approccio Low Code è rappresentato invece dalla modalità di svolgimento dell’attività di debug del codice; tale attività ha lo scopo di identificare e di correggere eventuali errori che si verificano durante il processo di sviluppo. A tale proposito la quasi totalità delle piattaforme di Low Code offre la possibilità di svolgere il debug in maniera semiautomatica evidenziando

allo sviluppatore gli errori, dove insorgono e perchè insorgono e fornendo quando possibile un suggerimento per la loro rapida risoluzione.

Alla luce di quanto delineato nel confronto tra le due metodologie di sviluppo risultano evidenti le differenze tra i due approcci. Lo sviluppo low-code offre un netto vantaggio rispetto allo sviluppo software tradizionale quando si tratta di creare prodotti minimi vitali (MVP) e/o di testare la fattibilità di un prodotto (POC). Questi approcci consentono di creare facilmente un prototipo di applicazione in tempi rapidi e con risorse relativamente limitate. [20]

Si opterà invece per un approccio di tipo tradizionale quando la soluzione da realizzare richiede un elevato grado di integrazioni complesse, funzioni aggiuntive e/o sofisticati protocolli di comunicazione e di sicurezza.

Nella figura sottostante viene riportata la statistica redatta da Mendix relativa ai casi d'uso che nell'anno 2021 hanno coinvolto lo sviluppo di software tramite piattaforme low code, mentre la figura 1.4 rappresenta le quote di mercato delle piattaforme Low code e no code nei vari settori.

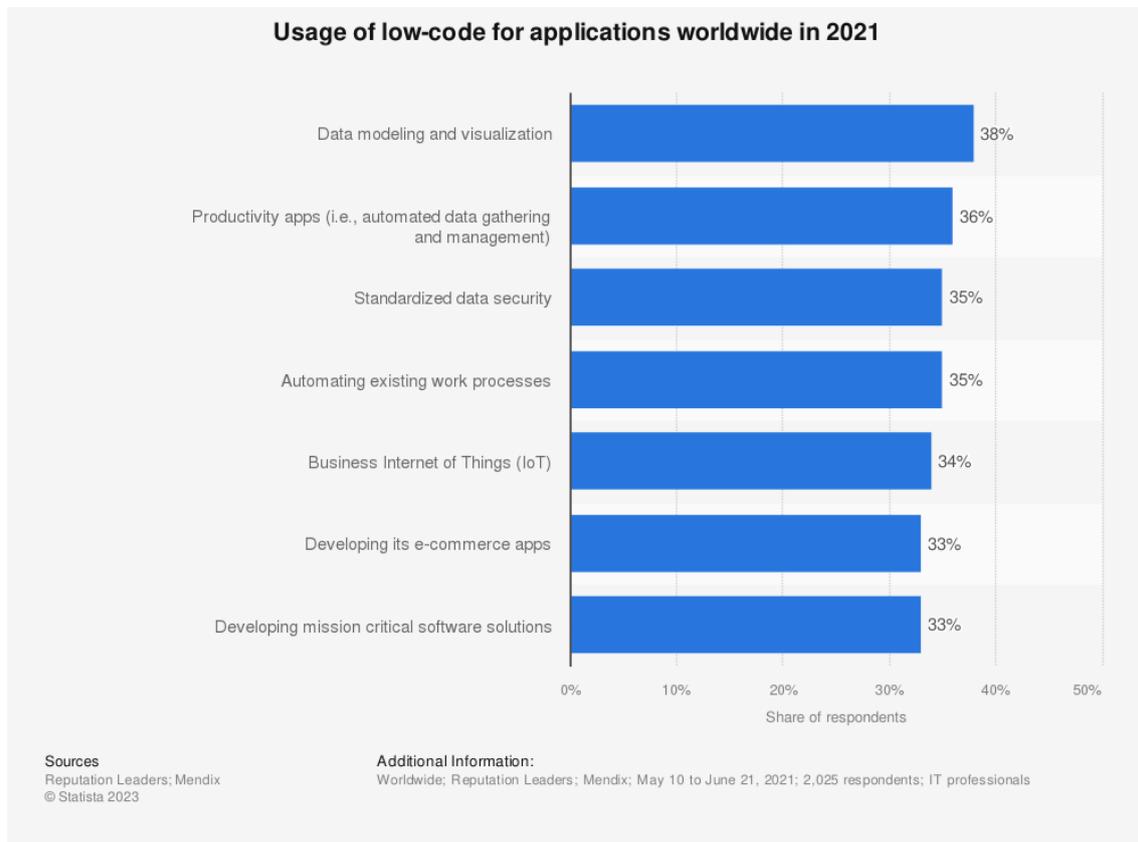


Figura 1.3. Utilizzo della metodologia low code per ambito. [2]

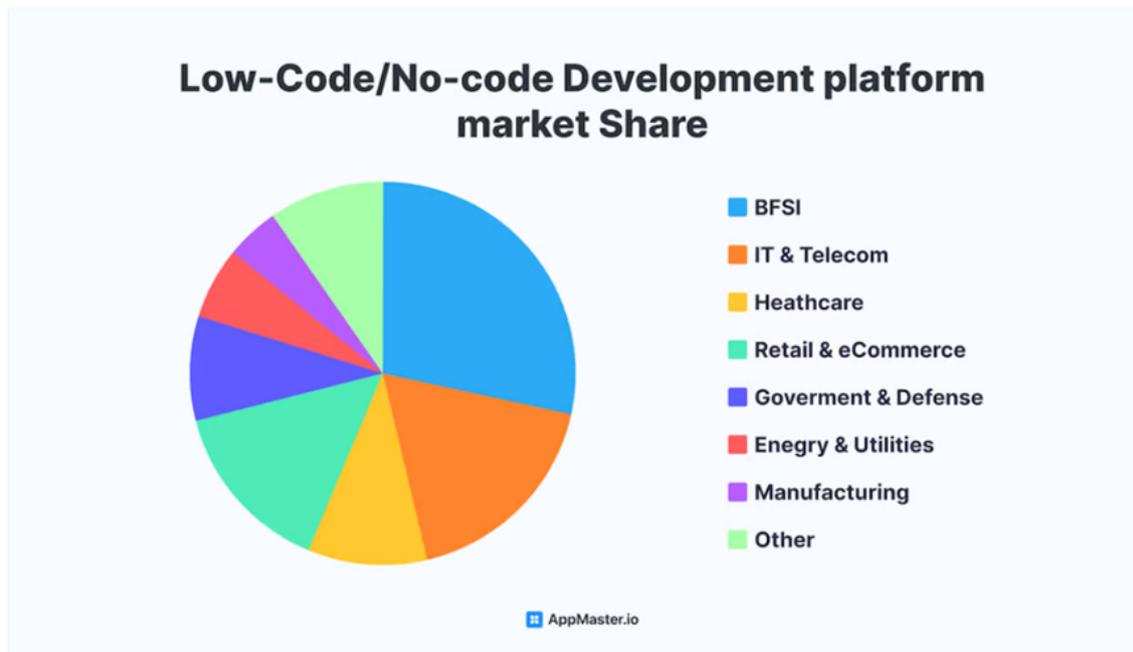


Figura 1.4. Utilizzo delle piattaforme Low Code/No Code per settore [3]

### 1.3 Approccio Data driven

L'approccio Data-driven, definito anche Data-centric, è un modello decisionale che prevede che i dati abbiano un ruolo centrale nel guidare le decisioni strategiche e operative dell'azienda.

Data-driven significa infatti essere “guidati dai dati”, ovvero utilizzare i dati e le informazioni che ne derivano per migliorare i processi decisionali, identificare problemi, effettuare analisi sulle quali impostare le proprie strategie.

Nel marketing, per esempio, l'analisi dei dati dei clienti (Customer Analytics) è ormai una prassi consolidata in diverse aziende perseguendo un duplice obiettivo: costruire relazioni più significative e durature con i clienti e anticipare le loro esigenze. [21]

Questo approccio metodologico prevede una costante e continua consultazione dei

dati disponibili al fine di automatizzare e/o rendere più efficienti i processi aziendali, di rendere più attraente la propria offerta, di affinare le strategie di marketing ecc. Di conseguenza, diventa fondamentale per il management avere a disposizione dati corretti, pertinenti, freschi e rilevati con frequenza. L'approccio data-driven applicato al processo decisionale prevede innanzitutto che vengano esplicitati gli obiettivi aziendali che si intendono perseguire, sia a livello di direzione che dei singoli reparti. Sono gli obiettivi, infatti, a definire i dati sui quali occorre focalizzare l'attenzione.

Una volta individuati i dati da raccogliere e analizzare occorre verificare se gli stessi sono già disponibili in azienda o se occorre fare ricorso a fonti esterne verificando i costi di acquisizione al fine di valutare se i benefici attesi superano i costi di acquisizione.

L'ambito nel quale l'utilizzo di un approccio data-driven è oggi più rilevante è il Marketing, dove risulta sempre più importante prevedere i comportamenti dei clienti e aumentare le vendite personalizzando le proposte. [22]

L'adozione di un modello data-driven parte dall'individuazione dei dati significativi nell'ambito applicativo di interesse. Nel Marketing, ad esempio, risultano particolarmente interessanti i dati presenti nel CRM (Customer Relationship Management), che contiene le informazioni relative ai clienti. L'integrazione di queste informazioni con dati di terze parti, soprattutto quelli provenienti dai social media, possono consentire ad una organizzazione di generare profili utente ed erogare contenuti in maniera ottimizzata. Anche in ambito HR il data-driven decision making si sta distinguendo come approccio in grado di apportare innovazione. I dati che la Direzione HR potrebbe trasformare in informazioni utili per attivare un processo decisionale più efficace sono ad esempio quelli di performance, i tassi di retention e turnover, i contributi provenienti dai social media, i risultati delle survey di clima, ecc. [23]

Volendo sintetizzare i benefici che un'azienda può trarre dall'utilizzo di un modello

Data-driven è possibile in prima battuta individuare un miglioramento dell'efficienza operativa in quanto l'analisi dei dati può portare all'individuazione di inefficienze come, ad esempio, l'improduttività di una determinata business unit o un elevato costo operativo sostenuto per un particolare progetto. Dal punto di vista del rapporto con la clientela è facilmente intuibile che un approccio data-driven, favorendo un maggior grado di personalizzazione delle soluzioni proposte, consentendo di segmentare al meglio la clientela e migliorando la customer experience, può tradursi in un notevole vantaggio competitivo per l'azienda.

L'approccio data-driven presenta anche delle criticità, dovute principalmente al corretto utilizzo dei dati ed alla tutela della privacy nel rispetto delle normative vigenti (ad esempio il GDPR). A tal proposito, il legislatore all'articolo 5 del GDPR precisa che: *“i dati devono essere adeguati, pertinenti e limitati a quanto necessario rispetto alle finalità per le quali sono trattati”* cioè: [24]

- devono essere in relazione con le finalità per cui sono raccolti;
- devono essere sufficienti e adeguati allo scopo del trattamento;
- devono essere necessari, non eccedenti e proporzionali;
- devono essere esatti;
- devono essere conservati in una forma che consenta l'identificazione degli interessati per un arco di tempo non superiore al conseguimento delle finalità per le quali sono trattati.

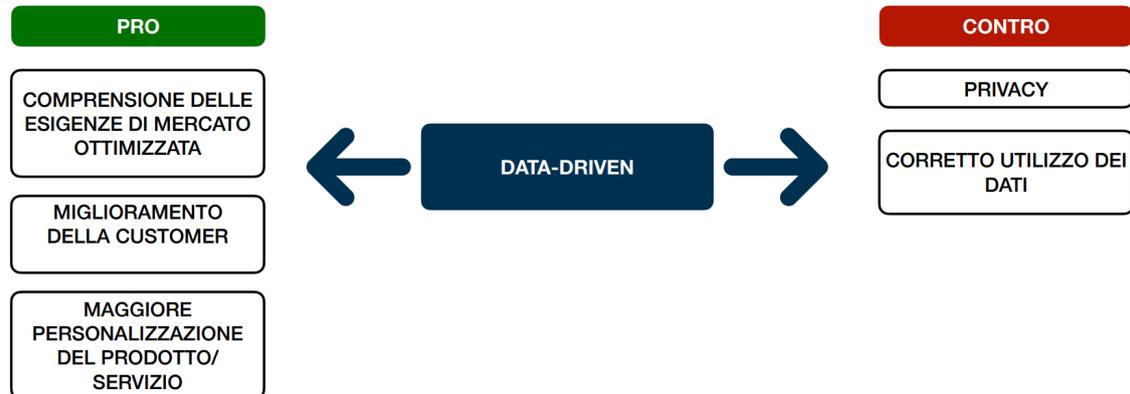


Figura 1.5. Caratteristiche approccio data-driven.

### 1.3.1 Complementarietà dell'approccio data-driven e dello sviluppo low-code

Alla luce di quanto detto in precedenza è possibile vedere gli approcci Data-driven e low code development come complementari che, se integrati correttamente hanno lo scopo di migliorare l'efficienza nello sviluppo di applicazioni e costituiscono uno strumento utile per le aziende in quanto sono portatori di vantaggio competitivo all'interno del settore dello sviluppo software e della consulenza IT.



Figura 1.6. Complementarietà degli approcci.

Nell'immagine sopra inserita è possibile visualizzare i punti di contatto dei due differenti approcci, partendo dall'alto è possibile fare alcune considerazioni a riguardo:

- **Integrazione dei dati:** spesso le piattaforme di sviluppo low code offrono funzionalità di integrazione delle informazioni, consentendo di manipolare i dati indipendentemente dalla loro forma e dalla loro rappresentazione e garantendo la possibilità di connettersi a più fonti di dati. Questo è un aspetto fondamentale per l'implementazione dell'approccio data-centric poiché la gestione in un unico ambiente di dati eterogenei provenienti da fonti diverse ne facilita l'accesso e l'utilizzo.
- **Accesso semplificato:** questo aspetto è collegato come definito sopra all'integrazione dei dati e alla possibilità di accedervi mediante semplici schermate.

- **Visualizzazione:** lo sviluppo low code permette di creare agevolmente dashboard, talvolta molto dettagliate per la visualizzazione dei dati consentendone una analisi rapida e semplificata.
- **Vicinanza al mercato:** Questo aspetto coinvolge entrambi gli approcci in quanto con una metodologia data-driven è possibile sfruttare i dati per capire le esigenze del mercato mentre con un approccio low code è possibile servire queste esigenze con minimo sforzo e minime modifiche al prodotto che, si traducono in minori costi per l'azienda.
- **Ottimizzazione:** a questo aspetto contribuiscono entrambi gli approcci anche se in momenti differenti. L'approccio Data-driven permette, tramite l'analisi dei dati, l'individuazione delle aree e/o dei processi inefficienti mentre l'approccio low code si colloca in un momento successivo all'individuazione del problema cioè nella fase di reazione consentendo di apportare al sistema le modifiche necessarie in modo rapido e mirato.



## Capitolo 2

# Panoramica sul mercato delle piattaforme di sviluppo low code

### 2.1 Mercato delle piattaforme di sviluppo Low Code.

A partire dal 2018, il settore delle piattaforme di sviluppo low code ha vissuto un periodo di forte crescita; sono infatti sempre di più le aziende che per le proprie applicazioni informatiche adottano soluzioni basate su queste piattaforme e si prevede che nei prossimi anni il mercato di questi strumenti per lo sviluppo software sia destinato ad aumentare ulteriormente.

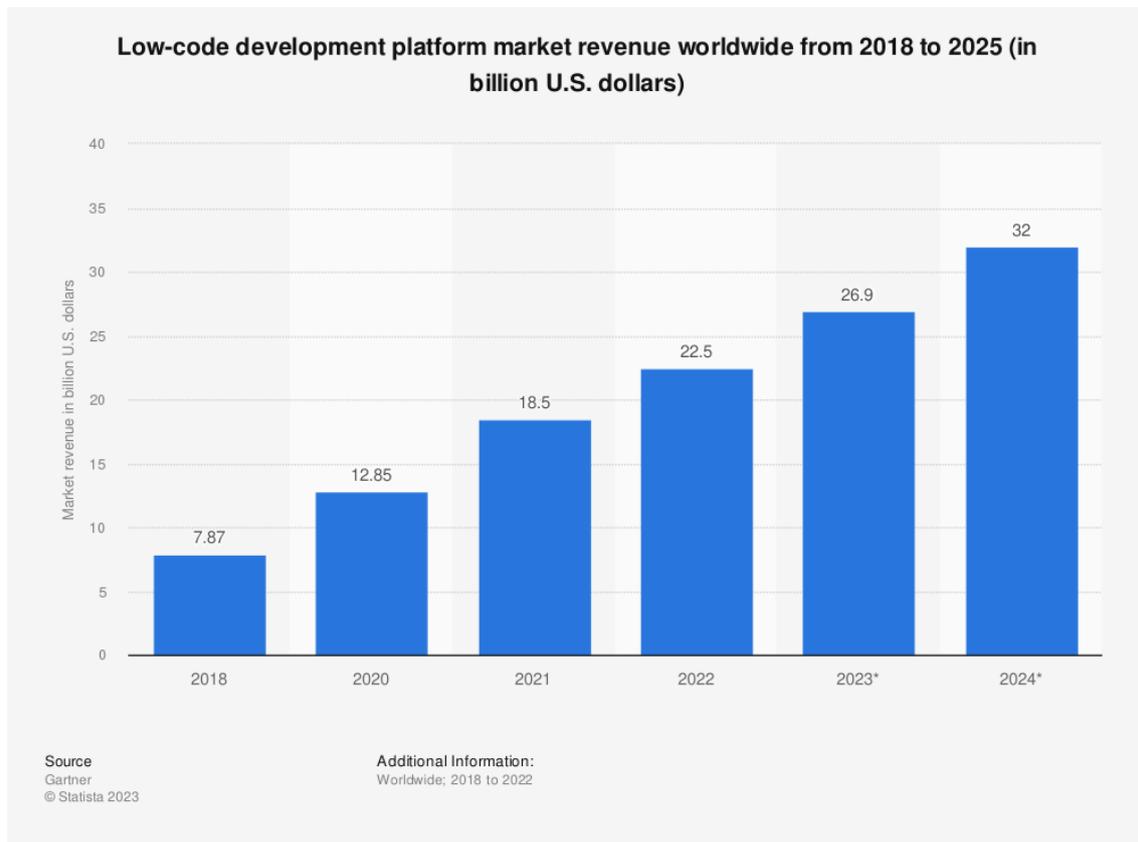


Figura 2.1. Revenues delle piattaforme di sviluppo low code [4]

Come è possibile notare consultando la figura sopra inserita, il mercato delle piattaforme di sviluppo low code ha generato dal 2018 revenues sempre crescenti a livello mondiale raggiungendo quasi 23 miliardi di dollari nel 2022; le previsioni sono incoraggianti e indicano un mercato ancora in crescita nei prossimi anni generando entrate globali per circa 65 miliardi di dollari nel 2027 con un Compound Annual Growth Rate (CAGR) stimato pari al 26,1 %.

La suddetta crescita del mercato delle piattaforme Low Code non sorprende più di tanto se si va ad indagare il pensiero degli sviluppatori cioè degli attori coinvolti in prima persona con le tematiche del settore. I feedback positivi di questi ultimi verso le piattaforme di sviluppo low code risultano infatti crescenti di anno in

anno; tuttavia, andando ad analizzare la percezione degli sviluppatori sulla qualità del software ottenuto con la metodologia Low Code riferita all'anno 2021, risulta evidente che i fornitori leader delle suddette piattaforme di sviluppo abbiano ancora diversi aspetti da migliorare per incrementare i livelli di adozione di questi strumenti per la realizzazione di applicazioni informatiche.

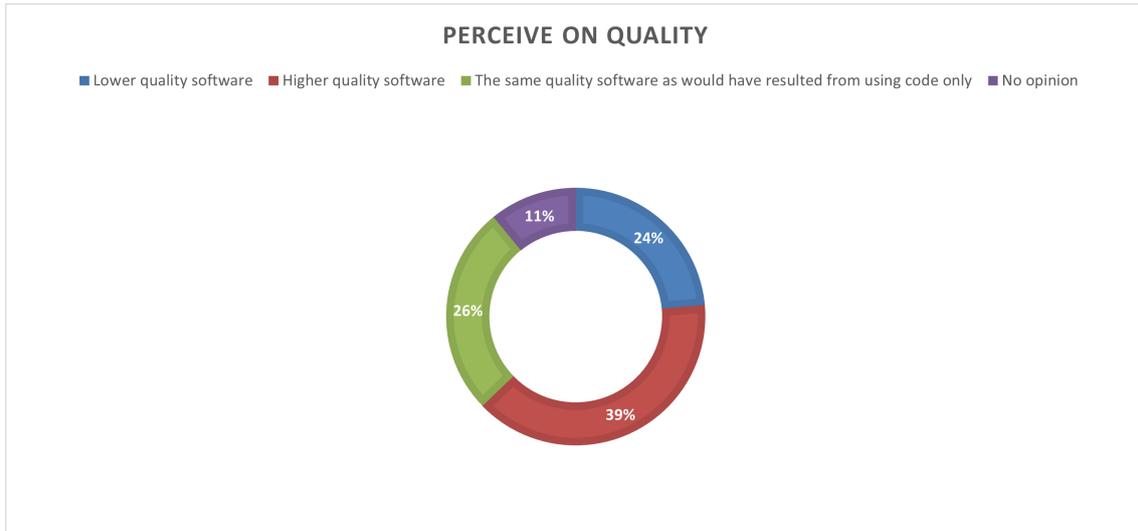


Figura 2.2. Qualità percepita dagli sviluppatori nel 2021. [5]

## **2.2 Attori del mercato**

Al fine di analizzare il posizionamento competitivo dei fornitori di tecnologia Low Code e di offrire alle aziende una guida alla selezione della piattaforma di sviluppo LCAP più adatta alle proprie esigenze, la società di consulenza IT Gartner ricapitola periodicamente in un report, il cosiddetto Magic Quadrant, tutte le principali aziende fornitrici di tale servizio. Il Magic Quadrant di Gartner è il risultato di una serie di analisi su dati qualitativi dei vari player tecnologici aventi lo scopo di evidenziare le tendenze del mercato in termini di direzione, maturità e partecipanti. Il rapporto, che può essere utilizzato come punto di partenza per la ricerca da parte di un'azienda di un nuovo fornitore di servizi nell'ambito IT, si basa su queste tre caratteristiche:

- Analisi dei trend di settore e del mercato.
- Valutazione del livello di maturità delle soluzioni proposte dai vari player.
- Capacità del fornitore di rispondere alle esigenze di servizi in ambito IT espresse dalle aziende e agli obiettivi di queste ultime.



Figura 2.3. Gartner low code platforms. [6]

Come si può osservare dalla figura, un Magic Quadrant di Gartner è basato sulle due variabili:

- **Completeness of vision** che si riferisce alla capacità del fornitore di servizi di anticipare o addirittura influenzare i trend di mercato in quello specifico settore.
- **Ability to execute** che analizza la capacità del fornitore di implementare con successo le proprie tecnologie nell'azienda cliente.

I quattro quadranti che ne derivano corrispondono a differenti tipologie di fornitore e i vari players presenti sul mercato vengono ricondotti alla tipologia che più li rappresenta mediante la distinzione tra: [25]

- ***Niche players:*** sono spesso concentrati su specifici segmenti di mercato; hanno prodotti interessanti con funzionalità uniche per certi settori o particolari aree geografiche. In genere sono in grado di offrire soluzioni complete su determinate tematiche ma mostrano debolezze in segmenti o aree di mercato di ampia scala.
- ***Visionaries:*** sono fornitori che puntano a prodotti innovativi per affrontare problemi di una certa portata per gli utenti finali, pur non avendo ancora acquisito quote di mercato significative in termini di reale soddisfacimento delle esigenze delle aziende. Solitamente le soluzioni offerte da questa tipologia di fornitori non sono ancora completamente ingegnerizzate e consolidate e, di conseguenza, espongono al rischio che l'investimento effettuato non si traduca nel beneficio atteso.
- ***Challengers:*** sono attori sufficientemente presenti sul mercato tanto da rappresentare una potenziale minaccia per i fornitori Leaders. Dimostrano forza nell'Ability to execute potendo contare su prodotti consolidati e avendo mostrato un ottimo operato nelle rispettive aree di interesse nonché capacità di espansione della propria base di clienti, pur non avendo ancora dimostrato una visione di ampliamento della loro offerta.
- ***Leaders:*** posseggono una buona quota di mercato e sono in grado promuovere l'accettazione e la diffusione delle nuove tecnologie LCAP grazie soprattutto alla loro credibilità e alle capacità di marketing. Questi fornitori dimostrano una chiara comprensione delle esigenze delle aziende coprendo con le loro applicazioni un'ampia gamma di organizzazioni e di casi d'uso.

Partendo dal Magic Quadrant di Gartner possiamo fare una breve carrellata delle principali piattaforme Low-Code attualmente presenti sul mercato esaminando sinteticamente i punti di forza e di debolezza di ciascuna di esse: [15] [26]

- **Mendix:** è una filiale dell'azienda Siemens e offre la piattaforma Mendix Platform. I suoi clienti sono distribuiti in imprese di tutte le dimensioni, soprattutto nel settore finanziario, dei servizi professionali e del manifatturiero. Secondo le analisi di Gartner uno dei punti di forza di questa piattaforma è l'innovazione. L'azienda fornisce infatti funzionalità sempre più avanzate di low-code per le imprese con continui nuovi casi d'uso, come l'Internet of Things (IoT) e i digital twins.

Una delle caratteristiche principali di Mendix è il suo ambiente di sviluppo che rende disponibile un'ampia gamma di elementi visivi e di modelli predefiniti. Grazie a questo ambiente è possibile ottenere una rappresentazione grafica dell'architettura dell'applicazione che consente agli utenti di gestire le relazioni, anche complesse, tra i diversi componenti.

Il prodotto di Mendix si distingue per l'elevato livello delle funzionalità chiave quali ad esempio il cosiddetto UX Desing cioè la progettazione degli oggetti in ottica di miglioramento dell'esperienza dell'utente. Offre vari editor che semplificano la collaborazione tra team diversi di sviluppatori non professionisti anche in caso di applicazioni aziendali complesse e una strategia di rilascio con compatibilità con le versioni precedenti che contribuisce a garantire scalabilità e affidabilità ai suoi clienti.

Uno dei punti deboli della soluzione secondo Gartner è la distribuzione geografica, in quanto la soluzione proposta da Siemens non si sta per ora diffondendo al di fuori degli Stati Uniti e dell'Europa. Altro elemento da superare è quello dei "verticali"; la piattaforma viene ad oggi prevalentemente utilizzata in ambito finance e manufacturing e dovrebbe puntare ad acquistare casi d'uso anche al di fuori di questi specifici settori.

- **Microsoft:** offre la piattaforma Power Apps, che comprende il servizio Dataverse per l'archiviazione e la gestione dei dati, il prodotto Power Pages per la creazione di siti Web low-code mediante l'utilizzo di modelli predefiniti e Power Automate per l'automazione delle attività ricorrenti. La piattaforma Power Apps è facilmente integrabile con tutti i prodotti dell'ecosistema Microsoft come Office, Dynamics, il cloud Azure e soprattutto "Power BI" che consente la creazione di dashboard per un'agile analisi dei dati. Questa caratteristica consente alle aziende di creare applicazioni che si allineano perfettamente con la infrastruttura software esistente. Secondo le analisi di Gartner le operations di Microsoft sono diversificate a livello geografico e i clienti appartengono a tutti i settori e a tutte le dimensioni aziendali. Come punto a sfavore di Microsoft Power Apps le analisi di Gartner evidenziano la politica delle licenze in quanto gli sviluppatori professionisti che vogliono estendere le funzionalità ad altre componenti di Microsoft si trovano a dover acquistare licenze separate. Un altro fattore di criticità è rappresentato dal fatto che Power App non è di immediato utilizzo per lo sviluppo di soluzioni per il mondo mobile.
- **OutSystems:** la piattaforma di OutSystems offre un ambiente di sviluppo low-code in grado di supportare applicazioni e componenti di qualsiasi complessità aumentando la produttività degli sviluppatori professionisti. La soluzione risulta sufficientemente attrezzata sotto il profilo della sicurezza e consente uno sviluppo e distribuzione delle applicazioni facilitato, grazie anche alla presenza di numerose utilities basate sull'intelligenza artificiale. Dispone di un proprio editor di interfaccia utente che consente una buona personalizzazione pur fornendo vari layout precostruiti e si caratterizza per la sua capacità di sviluppare applicazioni ottimizzate per i dispositivi mobili. Una caratteristica sicuramente interessante della piattaforma OutSystems è la possibilità di integrare le applicazioni con i sistemi esistenti.

Per quelle aziende dotate di sistemi legacy che non possono essere abbandonati, la possibilità di creare nuove applicazioni in grado di interagire perfettamente con i sistemi esistenti rappresenta un aspetto essenziale in quanto diventa fondamentale garantire la continuità e la coerenza dell'infrastruttura complessiva dell'azienda. Infine, la società offre un'offerta diversificata sui settori bancario, finanziario, assicurativo e dei servizi professionali.

- **Salesforce:** la Salesforce platform, è una suite per lo sviluppo convenzionale e low code che contempla strumenti di visual design “no-code” come “Lightning App Builder” e “Flow Builder” e un linguaggio di programmazione fortemente tipizzato e orientato agli oggetti come Salesforce Apex language. E' una delle piattaforme più diffuse nel mondo: i suoi clienti sono spesso aziende medio-grandi che utilizzano già prodotti CRM della stessa Salesforce. Uno dei principali punti di forza di Salesforce è la politica di rilascio delle nuove versioni: le nuove funzionalità della piattaforma vengono generalmente rilasciate in anteprima per gli sviluppatori e vengono pianificate tre release annuali di nuove funzionalità oltre alla distribuzione di patch durante tutto l'anno.

Il principale punto di debolezza individuato da Gartner sulla piattaforma Salesforce è rappresentato da un certo ritardo nell'innovazione in quanto attualmente non offre la possibilità di uno sviluppo integrato con l'intelligenza artificiale. Un'altra debolezza del prodotto è rappresentata dalla politica commerciale spesso basata su offerte a consumo diversificate su un gran numero di opzioni di acquisto.

- **Appian:** la piattaforma low Code di Appian si caratterizza per l'elevata scalabilità che consente di mantenere adeguati livelli prestazionali anche a fronte di un aumento significativo dei carichi di lavoro. La Appian Platform fornisce

una funzionalità per la modellazione dei processi mediante la quale è possibile progettare, visualizzare e gestire flussi di processi complessi tracciando il percorso delle informazioni al fine di identificare eventuali colli di bottiglia e impostare azioni di miglioramento.

La piattaforma offre inoltre una interessante funzionalità di analisi in tempo reale del funzionamento dei vari processi fornendo alcuni indicatori di performance. Il principale punto di debolezza segnalato dagli sviluppatori è la User Experience Design che dovrebbe essere potenziata per offrire funzionalità in grado di personalizzare maggiormente lo sviluppo orientato all'esperienza utente.

- **Oracle:** la piattaforma di sviluppo low-code APEX (Application Express) proposta dal colosso americano Oracle si distingue per la flessibilità in quanto si tratta di uno strumento che si può fruire a più livelli, da una base gratuita "Oracle FreeTier" fino a un approccio enterprise. E' uno strumento di classe enterprise che offre grande agilità per realizzare applicazioni scalabili e stabili nel tempo, facilmente aggiornabili e disponibili sia in Cloud che on-premises. Nelle applicazioni realizzate con Oracle APEX tutta l'elaborazione, la manipolazione dei dati e la logica di business vengono eseguite nel database Oracle garantendo così massime prestazioni e zero latenza. Oltre ad aspetti puramente tecnici e di struttura, APEX presenta come ulteriore punto di forza la capacità di generare applicativi fully-responsive. Questo è un enorme vantaggio competitivo in quanto è sempre più importante poter usufruire delle applicazioni da Web, Mobile e Tablet senza vedere la propria applicazione perdere metriche e proporzioni di riferimento. Anche dal punto di vista dei costi APEX risulta estremamente competitivo in quanto il prodotto è compreso nella licenza di acquisto di ogni Database Oracle ed è pronto per essere installato on-premises oppure per essere gestito interamente su Cloud. [27]

## 2.3 Sfide che il settore dovrà affrontare

In questi ultimi anni il modo di fare business delle imprese è sempre più influenzato da elementi quali l'imprevedibilità e la disruption, fattori che hanno determinato una forte richiesta di flessibilità nelle organizzazioni e nelle tecnologie da esse utilizzate. In risposta a questa necessità in ambito organizzativo e di disegno dei processi sono sempre più utilizzati modelli di tipo "Agile" mentre a livello tecnologico si fa sempre più ricorso a tecniche di "assemblaggio" che garantiscano l'integrabilità di semilavorati. [28]

Come si è visto nei capitoli precedenti, nell'ambito dello sviluppo applicativo le piattaforme Low Code che integrano un'ampia serie di componenti software semilavorati facilmente assemblabili e riutilizzabili stanno raggiungendo una fase di maturità tecnologica e cominciano ad essere utilizzate anche da aziende caratterizzate da una minore tolleranza alla perdita di continuità nelle applicazioni mission critical. Per superare le attuali diffidenze e conquistare un mercato globale, le tecnologie di sviluppo Low Code devono affrontare nel prossimo futuro alcune importanti sfide, prima fra tutte quella della sicurezza; queste piattaforme potrebbero infatti essere esposte a rischi per la sicurezza quali, ad esempio, l'esposizione dei dati e la compromissione del sistema e/o degli account.

Particolarmente delicato è l'aspetto della sicurezza dell'ambiente operativo che è l'ambiente nel quale le applicazioni vengono sviluppate e generalmente anche eseguite; su questo fronte non tutte le piattaforme disponibili dispongono di un ambiente di sviluppo dotato di adeguati strumenti di sicurezza quali ad esempio la crittografia, la identity federation e il logging. [29]

Un altro problema tipico delle applicazioni low-code è il rischio di esposizione dei dati; la maggior parte dei provider presenti sul mercato non offre funzionalità di data masking e per mitigare questo rischio è consigliabile che le applicazioni low-code siano posizionate dietro un sistema di intermediazione della sicurezza in grado

di offrire sia il monitoraggio dei dati sia i controlli di protezione delle informazioni in transito.

Per conquistare nuove quote di mercato ed intercettare le esigenze del futuro i fornitori di piattaforme Low code dovranno far fronte alla crescente richiesta di integrazione dell'Intelligenza Artificiale e dell'apprendimento automatico (Machine Learning) semplificando il processo di incorporamento di queste tecnologie avanzate all'interno delle applicazioni aziendali.

Un ulteriore ambito di crescita per le piattaforme Low Code potrebbe essere quello dell'Internet delle cose (IoT) dove il Low-Code potrebbe rappresentare un'ottima soluzione per semplificare lo sviluppo, l'implementazione e la manutenzione delle applicazioni che vengono realizzate per governare i nuovi ambienti interconnessi.

E' infine auspicabile che l'espansione del mercato delle piattaforme Low Code sia accompagnata dalla definizione di standard e certificazioni relative a questa nuova tecnologia; attraverso un processo di progressiva standardizzazione sarà infatti possibile garantire l'interoperabilità delle soluzioni low-code, offrire una migliore qualità delle stesse e aumentare la fiducia delle aziende verso questa nuova modalità di sviluppo software. In merito alle evoluzioni future del settore Low Code, una recente analisi di Gartner sulle tecnologie emergenti (pubblicata nel luglio 2022) indica che dal 2027 almeno il 50% degli investimenti dei fornitori di piattaforme low code riguarderanno l'integrazione di PBC, Packaged Business Capabilities, cioè semilavorati software con funzionalità di processo integrate focalizzate su specifiche aree e attività di business. [7]

Sempre secondo lo studio prospettico di Gartner, dopo una prima fase di democratizzazione delle tecnologie per lo sviluppo software, si passerà ad una fase (detta di Hyperautomation) il cui obiettivo consisterà nell'integrazione e armonizzazione di funzioni e processi usati nei differenti silos aziendali. In questa fase i produttori di piattaforme Low Code dovranno investire nell'architettura concentrandosi sull'automazione a livello di processo mediante l'integrazione di tecnologie RPA (Robotic

Process Automation), BPA (Business Process Automation), tecniche di Machine Learning e Intelligenza Artificiale (AI). [28]

I vari vendor dovranno garantire la possibilità di automatizzare processi multipli, di gestire ampi workflow legati a processi complessi e interdipendenti.

### Low-Code Market Opportunities

Market Subsegment	Legend		
	■ Significant Market Opportunities	■ Average Market Opportunities	■ Lacking Opportunities
	Phase 1: Augmented Data Analytics	Phase 2: Augmented Consumer	Phase 3: Generative Data and Analytics
Low-Code Application Platforms	Significant	Average	Significant
Business Process Automation Platforms	Average	Significant	Significant
Multiexperience Development Platforms	Significant	Average	Significant
Robotic Process Automation Platforms	Average	Significant	Average
Application Composition Platforms	Lacking	Significant	Significant
Other LCD Technologies*	Average	Lacking	Lacking

Source: Gartner

\* Other low-code development (LCD) technologies include rapid mobile app development (RMAD) tools and rapid application development (RAD) tools. RAD tools are on-premises only and mainly desktop-bound applications. Low code is the evolution of RAD to cloud and SaaS models.

754212\_C

**Gartner**

Figura 2.4. Opportunità di mercato lowcode entro il 2027 secondo Gartner [7]

In una terza fase di sviluppo delle future piattaforme Low Code si arriverà, secondo Gartner, al concetto di “composable enterprise”. A questo punto le piattaforme Low Code dovranno arricchirsi di funzionalità tali da consentire agli utenti una facile componibilità di semilavorati software per “costruire”, assemblando e riassemblendo rapidamente, nuove funzioni e processi in risposta alle più svariate esigenze di business (supporto decisionale, opportunità improvvise, disruption impreviste, ecc). [28] Gli elementi di business da comporre e orchestrare dovranno comprendere prodotti, servizi, modalità operative e organizzazione aziendale e potranno essere estesi anche ai partner e alle diverse operation.



# Capitolo 3

## Oracle APEX

Oracle APplication EXpress (APEX) è la piattaforma di sviluppo low code enterprise commercializzata da Oracle, società multinazionale del settore informatico; la sua prima versione è stata lanciata nel 2004 e viene aggiornata costantemente con l'implementazione di nuove funzionalità.

APEX consente lo sviluppo di applicazioni web basate su database Oracle (sia on premises che in Cloud) per la visualizzazione, manipolazione e conservazione dei dati utilizzando una semplice interfaccia web.

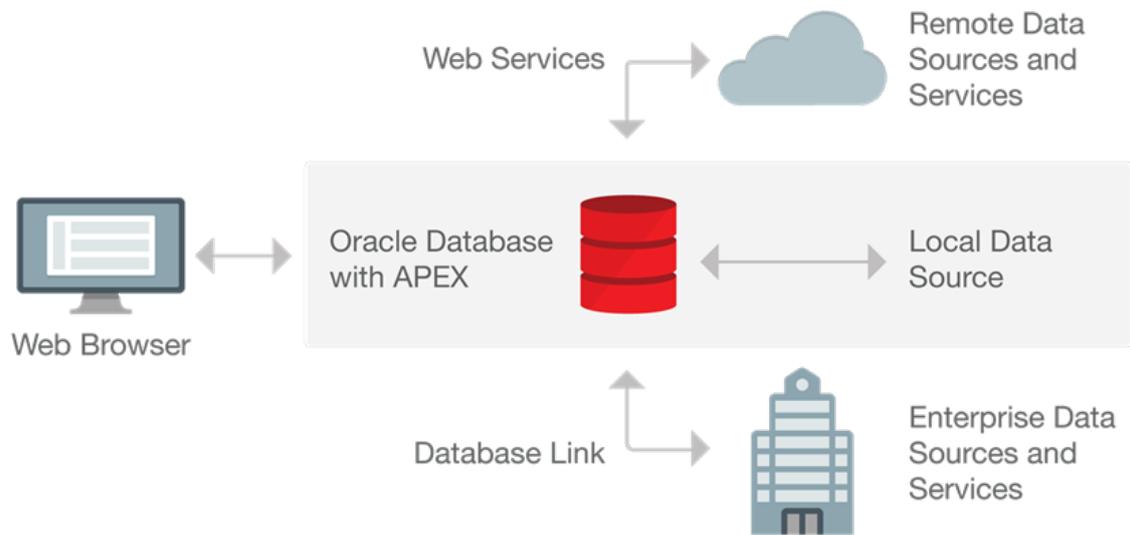


Figura 3.1. Architettura di un ambiente con Oracle Apex. [8]

Oracle APEX è un browser-based Rapid Application Development (RAD) tool che consente di sviluppare desktop e mobile enterprise web app in modo semplice e veloce pur richiedendo comunque alcune skill di SQL e conoscenze sui database. Le applicazioni web vengono realizzate ed eseguite su database Oracle garantendo sia l'accesso ai dati sia l'integrazione con il database in totale sicurezza; sia nella fase di sviluppo che nell'esecuzione delle applicazioni non è necessario ricorrere ad alcun client software. In un IDE (Integrated Development Environment) gli unici requisiti necessari all'utilizzo del tool sono un browser e la connessione ad una rete Internet o Intranet. Ogni dato dell'applicazione viene immagazzinato nelle tabelle del database Oracle come un metadato che può essere manipolato utilizzando query SQL o PL/SQL personalizzate.

## 3.1 Architettura Oracle Apex

Apex utilizza la tipica architettura software di una web application basata su “tre livelli” secondo il seguente schema:

- **Livello di presentazione:** si tratta tipicamente di una pagina o di un sito web e rappresenta la parte di applicazione visibile all’utente (la cosiddetta interfaccia utente); contiene sia i componenti definiti per presentare le informazioni agli utenti sia le funzionalità che consentono agli utenti di interagire con il sistema. Le pagine sono solitamente realizzate con linguaggi come HTML, Javascript, JSP e le richieste vengono inviate dal browser al database passando attraverso un Web server.
- **Livello applicativo:** rappresenta la parte invisibile all’utente e si occupa della logica elaborativa del sistema. E’ questa la componente applicativa che interpreta ed esegue le richieste dell’utente interrogando il database. Questo livello è solitamente sviluppato con linguaggi come Java, PHP, Javascript.
- **Livello dei dati:** rappresenta l’insieme di quei servizi che sono indipendenti dal Web, come ad esempio un gestore di database (MySQL, Oracle, ...), per la persistenza delle informazioni. Tipici esempi di applicazioni presenti a questo livello sono server di dati (DBMS) o i server di posta elettronica.

In un ambiente Oracle il Web server responsabile dell’indirizzamento delle richieste dal Browser al database è Oracle REST Data Service (ORDS) mentre sarà competenza esclusiva del sistema di gestione del database qualsiasi attività di processamento e manipolazione dei dati al fine di garantire un accesso ai dati rapido e sicuro con tempi di latenza pari a zero e conseguente aumento delle performance complessive.

Internamente al database Oracle le richieste sono elaborate dall’Oracle Apex Engine; una volta completata l’elaborazione il risultato verrà inviato al browser tramite

i servizi REST. Ogni volta che viene creata o modificata un'app, Apex crea o modifica i dati presenti nelle relative tables del database. Quando l'applicazione verrà eseguita, Apex leggerà i relativi metadati e sarà in grado di elaborare le richieste tramite SQL queries.

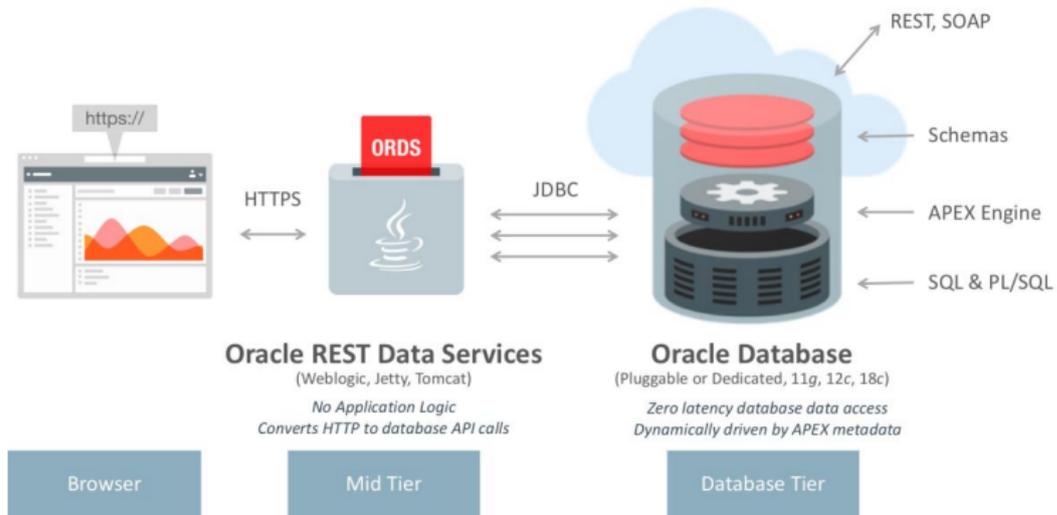


Figura 3.2. Architettura Oracle Apex. [9]

## 3.2 Oracle Application Express Development Environment

L'ambiente di sviluppo Integrato (IDE) di Oracle Apex è costituito da diversi componenti utilizzabili all'interno di un workspace precedentemente definito.

Un workspace è sostanzialmente un ambiente di lavoro virtuale all'interno del quale i diversi utenti, opportunamente autorizzati e identificati, possono operare. Un workspace funziona come un contenitore logico nel quale sono inseriti dati, oggetti ed applicazioni specifici per un progetto e all'interno del quale è possibile stabilire una gerarchia di ruoli, definendo così le attività che i vari utenti possono svolgere in base al ruolo che viene loro assegnato.

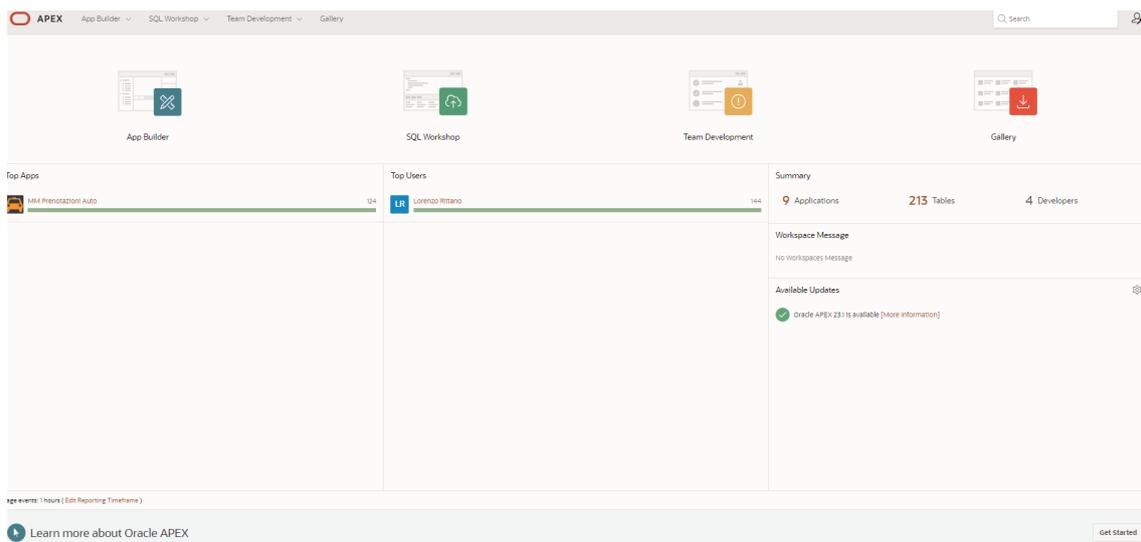


Figura 3.3. Workspace Oracle Apex

Oracle Application Express consente a un singolo database di ospitare un gran numero di applicazioni e utenti. Gli sviluppatori lavorano all'interno del workspace al quale sono stati abilitati. E' prassi comune definire spazi di lavoro separati per i diversi reparti all'interno di un'organizzazione in modo che ciascun reparto possa sviluppare i propri oggetti e applicazioni di database in modo indipendente come riportato in figura 3.4.

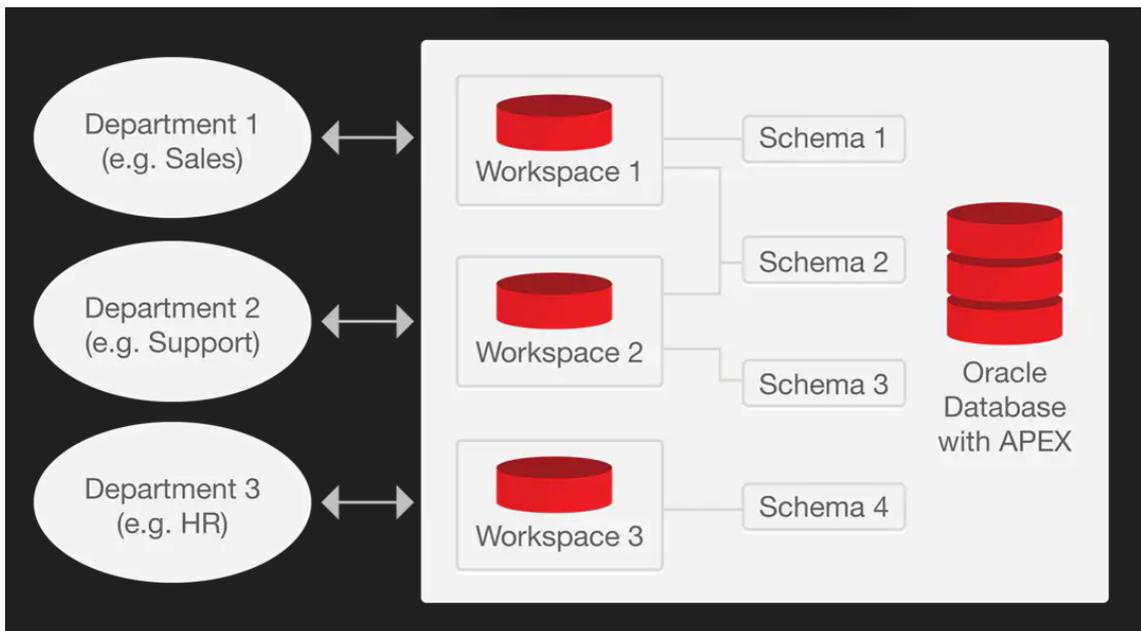


Figura 3.4. Workspace Oracle Apex [8]

Si può passare ora all'analisi dei vari componenti presenti all'interno di un workspace appositamente definito per un progetto.

### 3.2.1 Application builder

L'Application Builder mostrato in figura è il componente di Apex, presente all'interno di ogni workspace, che consente di creare o modificare applicazioni composte da un insieme di pagine HTML basate su oggetti di database e di importare applicazioni precedentemente esportate.

L'application builder è diviso in due sezioni:

- una prima sezione che dà accesso alle funzioni di creazione di una nuova app, all'import di applicazioni precedentemente esportate o sviluppate in un ambiente esterno, ad una dashboard contenente statistiche relative alle attività svolte all'interno del workspace e alle workspace utilities ovvero a funzionalità che consentono di accedere ai servizi SQL abilitati, ai server remoti o di organizzare i backup e i temi del workspace;
- una seconda sezione che mostra l'elenco delle applicazioni (rappresentate dalle relative icone) che fanno parte del workspace in uso; da questa sezione è possibile accedere ai dettagli della singola applicazione per andarla ad eseguire, modificare o esportare.

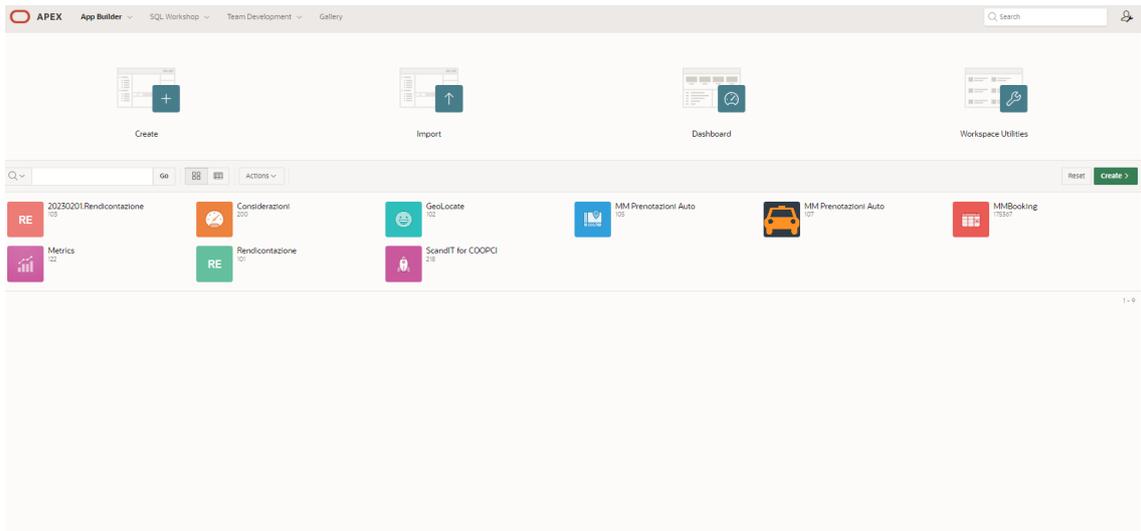


Figura 3.5. App Builder Oracle Apex.

### 3.2.2 SQL Workshop

Il SQL Workshop è il modulo Apex che consente, oltre alla scrittura e all'esecuzione di queries SQL e PL/SQL, la definizione degli oggetti che fanno parte dell'infrastruttura del DB come tabelle, viste, Triggers ecc.; il SQL Workshop permette infatti di creare rapidamente tali elementi, di visualizzarli, aggiornarli o eliminarli. Il SQL Workshop si compone di cinque moduli differenti come mostrato nella figura seguente:

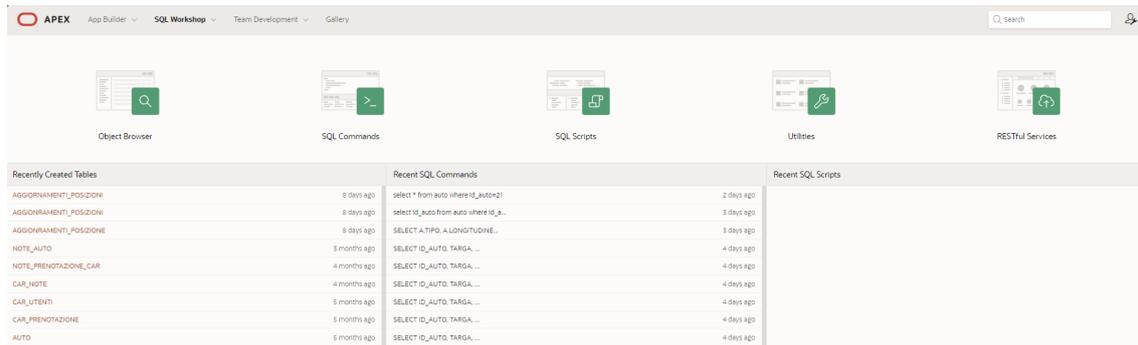


Figura 3.6. SQL workshop.

Nel dettaglio:

- **Object browser:** questo componente presenta una struttura ad albero per visualizzare le caratteristiche dei vari oggetti e per creare nuovi oggetti all'interno del database.

Come si può vedere dall'esempio riportato in figura 3.7, nella sezione di sinistra è presente l'elenco degli oggetti con gli elementi che li compongono; una volta selezionato un oggetto nella sezione di sinistra, l'object browser ne mostra le caratteristiche nella sezione di destra. Se, ad esempio, l'oggetto selezionato è una tabella, nella sezione di destra vengono mostrati i campi della tabella e tutte le caratteristiche che sono state implementate sulla tabella come indici, vincoli, triggers ecc.

The screenshot shows the Oracle APEX Object Browser interface. The 'Table' tab is selected, displaying the structure of the 'AUTO' table. The table has the following columns:

Column Name	Data Type	Nullable	Default	Primary Key
ID_AUTO	NUMBER	NO	-	1
TARGA	VARCHAR(27)	NO	-	-
MARCA	VARCHAR(255)	YES	-	-
TIPO	VARCHAR(255)	YES	-	-
UFFICIO	VARCHAR(255)	YES	-	-
CREATION_BY	VARCHAR(255)	YES	-	-
LAST_UPDATE_BY	VARCHAR(255)	YES	-	-
CREATION_DATE	DATE	YES	-	-
LAST_UPDATE_DATE	DATE	YES	-	-
LONGITUDINE	NUMBER	YES	-	-
LATTITUDINE	NUMBER	YES	-	-
DATA_REVISIONE	DATE	YES	-	-
LUOGO_REVISIONE	VARCHAR(255)	YES	-	-
TAGLIANDO	DATE	YES	-	-
KM_AL_TAGLIANDO	NUMBER	YES	-	-
AUTONOLEGGIO	VARCHAR(255)	YES	-	-
SCADENZA_ASSICURAZIONE	DATE	YES	-	-
INDIRIZZO_MANUTAZIONE	VARCHAR(255)	YES	-	-
INDIRIZZO_CAMBIO_COMME	VARCHAR(255)	YES	-	-

Figura 3.7. Object browser Oracle Apex.

- SQL commands:** è il componente di Apex che consente di visualizzare, inserire e modificare oggetti del database. Tramite questo componente è possibile eseguire queries SQL o blocchi PL/SQL visualizzandone o esportandone i risultati in fogli di calcolo. Lo strumento offre inoltre la possibilità di salvare queries per poterle in futuro integrare o eseguire nuovamente.

The screenshot shows the Oracle APEX SQL Commands interface. The 'SQL Commands' tab is selected, displaying a query and its results. The query is:

```

1 select 1
2 from car_prenotazione p, car_utenti u
3 where u.id_utente=p.id_utente and
4 u.username='P9999_USERNAME' and
5 to_char(sysdate, 'YYYYMMDDHH24') > to_char(p.data_inizio, 'YYYYMMDDHH24') and
6 trunc(sysdate, 'dd') = trunc(p.data_inizio, 'dd') and
7 p.stato_prenotazione=1;
    
```

The results table shows the following data:

Results	Explain	Describe	Saved SQL	History
4 4963 820			SELECT ID_AUTO,TARGA,MARCA,TIPO,UFFICIO,CREATION_BY	APEX_OSTICKET
4 4963 820			SELECT ID_AUTO,TARGA,MARCA,TIPO,UFFICIO,CREATION_BY	APEX_OSTICKET
4 4963 820			SELECT ID_AUTO,TARGA,MARCA,TIPO,UFFICIO,CREATION_BY	APEX_OSTICKET
4 4963 820			SELECT ID_AUTO,TARGA,MARCA,TIPO,UFFICIO,CREATION_BY	APEX_OSTICKET
4 4963 820			select ID_AUTO,TARGA,MARCA,TIPO,UFFICIO,CREATION_BY	APEX_OSTICKET
6 5863 820			SELECT * FROM APEX_USER,sysadm,PT_AGGIORNA_LATTITUDINE,PT_AGGIORNA_LONGITUDINE,COLESSIONE	APEX_OSTICKET

Figura 3.8. SQL commands.

- **SQL script:** è un modulo adibito alla gestione degli script SQL e PL/SQL; tramite questo modulo è possibile creare, modificare ed eseguire uno script SQL salvandolo direttamente all'interno del repository SQL Script. Mediante il repository di SQL Script è anche possibile caricare e scaricare scripts all'interno del filesystem locale.
- **SQL Utilities:** tramite il modulo Utilities Oracle Apex offre la possibilità allo sviluppatore di scaricare o caricare file in formato CSV, XLSX, XML, JSON. Questo modulo offre anche la possibilità di progettare agevolmente modelli di dati tramite la sintassi abbreviata del linguaggio SQL definito Quick SQL. Una delle caratteristiche maggiormente innovative della piattaforma Apex è proprio Quick SQL che consente allo sviluppatore di creare oggetti di un database fornendo in input istruzioni caratterizzate da una sintassi abbreviata. Nella seguente immagine è possibile vedere un esempio di tale funzionalità.

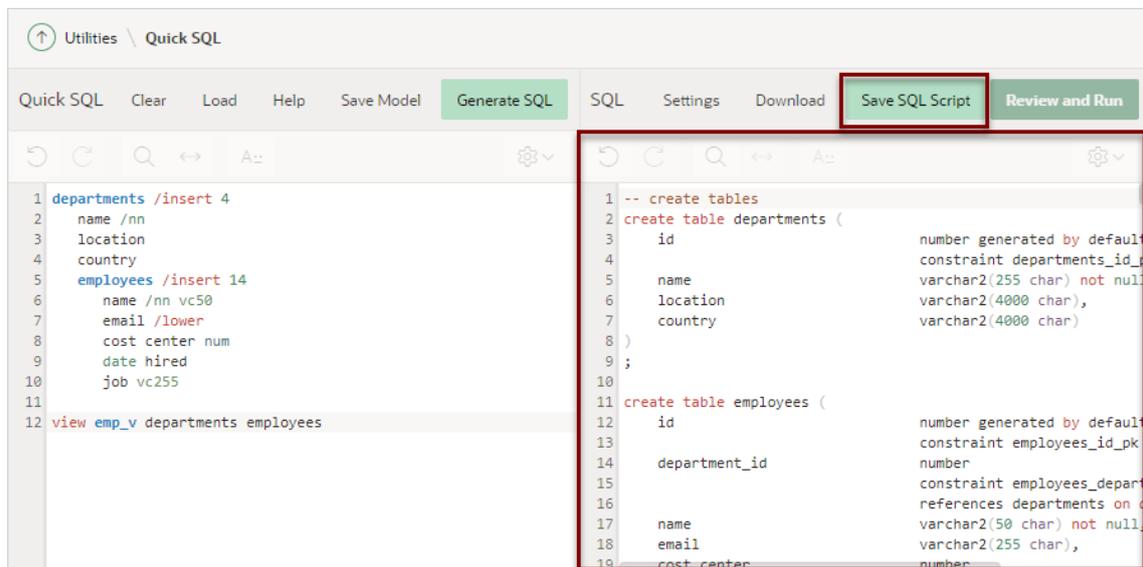


Figura 3.9. Esempio QuickSQL. [10]

- **RESTful Services** : il Representational State Transfer (REST) è un modo per fornire interoperabilità tra sistemi informatici su Internet. È possibile definire i servizi RESTful per consentire l'esecuzione di query e la manipolazione dei dati senza la necessità di accedere direttamente all'archivio dati sottostante. I RESTful Services presenti sulla piattaforma Apex consentono la creazione di una serie di API standard-based sugli oggetti di database disponibili all'interno di un workplace. Senza connettersi direttamente all'origine dati sottostante, queste API consentono ai sistemi esterni di interagire in modo sicuro eseguendo query, inserendo, aggiornando o eliminando dati. Le definizioni dei servizi RESTful create in Oracle APEX vengono archiviate nel repository Oracle REST Data Services (ORDS) e vengono definite ORDS-based REST Services.

### 3.2.3 Team development

Questa sezione serve per la gestione del Team di sviluppatori che sta lavorando su un progetto. E' possibile comunicare all'interno del Team per risolvere problemi, assegnare la risoluzione di eventuali problematiche ai vari membri del gruppo e impostare delle scadenze di progetto.

### 3.2.4 Gallery

Questa sezione del workspace offre allo sviluppatore una serie di applicazioni "per iniziare" e applicazioni "di esempio" che assolvono a funzionalità specifiche e che lo sviluppatore potrà consultare per impostare i propri progetti di sviluppo o trovare una soluzione ad un problema sorto durante lo sviluppo. La Gallery di Oracle Apex contiene anche una serie di app completamente funzionanti che forniscono soluzioni autonome per requisiti semplici e che possono essere usati così come sono oltre che essere estese con funzionalità personalizzate.

## 3.3 Le applicazioni in Oracle Apex

Dopo aver preso confidenza con l'ambiente di lavoro finora descritto, lo sviluppatore potrà agevolmente creare una o più applicazioni all'interno del workspace al quale è stato abilitato.

Un'applicazione APEX viene realizzata mediante la creazione due o più pagine web, che vengono modellate e modificate all'interno del Page Designer cioè in un ambiente di sviluppo integrato (IDE – Integrated Development Environment) completamente equipaggiato. In questo ambiente, suddiviso in una o più aree chiamate Apex Regions, sono presenti gli items forniti da Apex (Page Items) e lo sviluppatore può inserire all'interno di ciascuna pagina delle Dynamic Actions per generare delle azioni quando si verifica una determinata condizione e dei processi (Page Processes) per eseguire delle azioni durante la visualizzazione e la sottomissione della pagina. L'anatomia di un'applicazione Oracle Apex può essere raffigurata dalla immagine seguente.

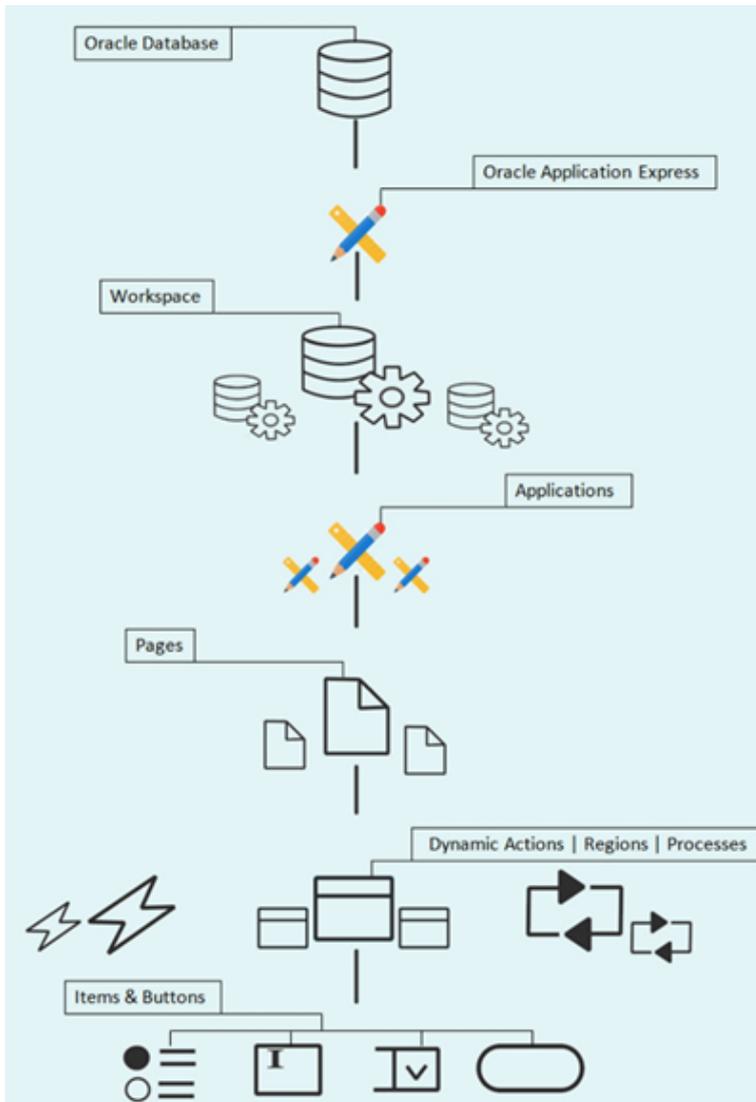


Figura 3.10. Anatomia applicazione Oracle Apex. [11]

Un'applicazione in Oracle APEX è quindi rappresentata da un insieme di pagine che possono essere collegate tra loro mediante un menu di navigazione, delle schede, dei pulsanti o dei links.

All'interno di una pagina, in sezioni chiamate "Regions", possono essere inseriti i diversi elementi dell'interfaccia utente utilizzando tutti i componenti disponibili in APEX come report, grafici, calendari, forms, pulsanti di elaborazione mediante

submission o indirizzamento ad altre pagine, ecc. All'interno delle pagine di un'app possono anche essere eseguiti processi di calcolo e di convalida.

Per visualizzare una pagina, APEX esegue due processi distinti:

- il processo Show Page unisce tutti i componenti di una pagina e li traduce in una pagina HTML visualizzabile;
- il processo Accept Page si occupa del processamento di una pagina effettuando calcoli, validazioni, eseguendo processi e branching.

### 3.3.1 Page Designer

Come detto poc'anzi, lo strumento di Oracle Apex mediante il quale lo sviluppatore definisce le pagine che comporranno l'applicazione e il loro comportamento è il Page Designer che permette di definire tutti gli elementi costitutivi di una pagina web ed è suddiviso in varie sezioni come mostrato in figura.

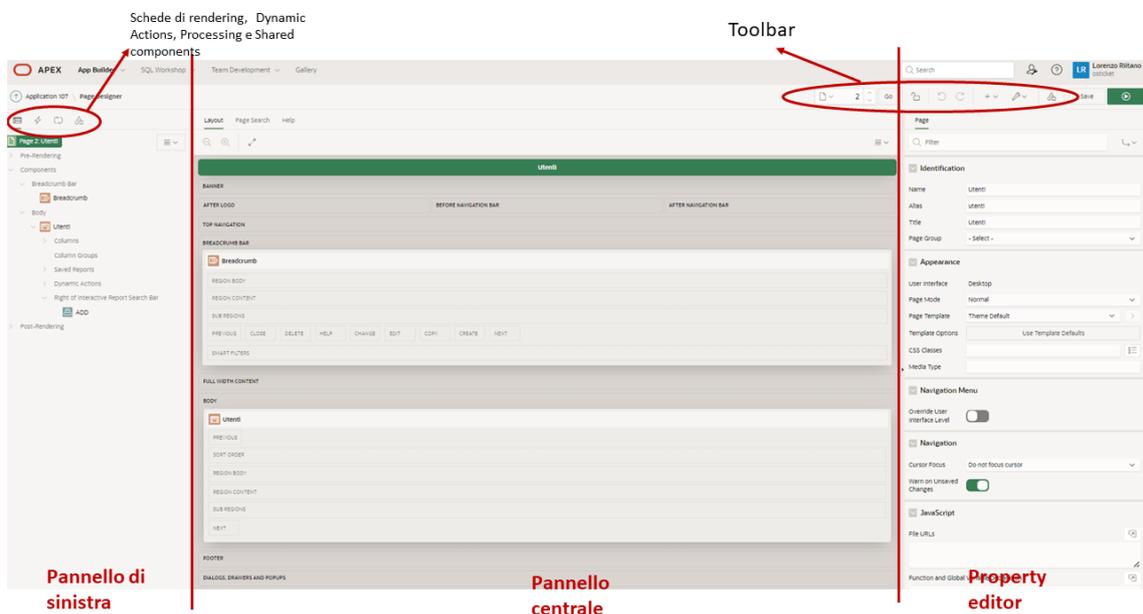


Figura 3.11. Page designer.

- **Una Toolbar:** che come tutte le toolbar presenta svariate funzionalità quali il salvataggio delle modifiche e il run della pagina; tramite questa sezione è anche possibile bloccare o sbloccare l'accesso alla pagina ad alcuni utenti. Una funzionalità interessante disponibile nella toolbar è quella relativa alla possibilità di collegarsi alla pagina degli Shared Components cioè alla pagina nella quale è presente una serie di elementi che potranno essere visualizzati su ogni pagina dell'applicazione.
- **Un left pane** la sezione laterale di sinistra, è la sezione di Page Designer nella quale vengono definiti, mediante una struttura a Tab gli elementi costitutivi di una pagina, i processi, le dynamics action e gli eventuali elementi condivisi con altre pagine del progetto. Questa è quindi la sezione nella quale viene definita la logica applicativa della pagina: con la funzionalità di Rendering vengono create le regioni di una pagina e vengono inseriti gli items e i bottoni; all'interno del tab Dynamic Action è possibile inserire nella pagina un comportamento complesso lato client senza l'utilizzo di JavaScript mentre la scheda Processing consente di creare e visualizzare la logica della pagina ordinando i componenti in base a come verranno processati da Apex. Infine, la funzionalità Shared Components consente di visualizzare ed inserire all'interno della pagina componenti condivise con altre pagine dell'applicazione.
- **Un Pannello centrale:** il pannello centrale di Page Designer è suddiviso a sua volta in due sezioni, una superiore contenente le schede navigabili di Layout, Page search e Help ed una inferiore definita Gallery. La scheda di Layout mostra un'anteprima degli elementi che costituiscono la pagina; è una rappresentazione grafica che visualizza la disposizione degli elementi all'interno della pagina. La sezione Gallery mostra l'elenco degli items inseribili all'interno della pagina ed è possibile spostare o aggiungere nuovi elementi

tramite semplici azioni di drag and drop trascinandoli dalla sezione inferiore al punto desiderato nella sezione superiore. Le schede page search e help sono funzionali rispettivamente alla ricerca rapida di tutti i metadati nella pagina ed alla spiegazione testuale dei vari items.

- **Property editor:** il pannello laterale destro, definito anche property editor mostra le informazioni relative ad un qualsiasi componente presente nella pagina (ad esempio una Region, un Page Item, un Button ecc.); una volta selezionato un elemento dal Left pane o direttamente dal Central pane nel Property editor saranno visibili e modificabili tutte le caratteristiche che definiscono tale componente quali le informazioni identificative, le modalità con cui l'elemento viene mostrato nella pagina e soprattutto la sorgente dei dati associati all'elemento, vale a dire se si tratta di dati prelevati direttamente da una tabella o se l'elemento rappresenta il risultato di una query.



# Capitolo 4

## MMPrenotazioni Auto

In questo capitolo viene illustrato il processo che ha portato allo sviluppo dell'applicazione web MMPrenotazioni Auto per la geolocalizzazione e la prenotazione della flotta auto aziendale e la gestione del relativo workflow approvativo. Vengono esplicitati i requisiti e le funzionalità discusse in fase di progettazione, viene illustrato il processo di sviluppo e viene fornita una descrizione dettagliata del funzionamento dell'applicazione.

### 4.0.1 Il contesto.

La società Mediamente Consulting Srl, azienda ospitante e sede di tirocinio, è una azienda con sede legale ad Empoli che offre servizi di consulenza IT con uffici dislocati presso le città di Torino, Bologna, Cagliari e Milano.

La natura consulenziale dell'azienda prevede che i dipendenti debbano svolgere brevi trasferte presso gli uffici delle aziende clienti; per favorire gli spostamenti in autonomia del personale dipendente la società Mediamente Consulting si è dotata di una flotta auto aziendale che necessita di una gestione coordinata e di un servizio per la prenotazione e geolocalizzazione delle auto aziendali.

## 4.0.2 Analisi dei requisiti.

L'applicazione Web MMPrenotazioni Auto, sviluppata durante il periodo di tirocinio, nasce dall'esigenza di sostituire il precedente modello di gestione e prenotazione della flotta auto aziendale basato sull'utilizzo di un foglio excel privo di qualsiasi workflow approvativo come riportato nella figura che segue.

Poiché l'azienda non dispone di parcheggi sufficienti ad ospitare tutte le auto in dotazione, per la localizzazione delle auto aziendali disponibili e il coordinamento dei dipendenti veniva utilizzato semplicemente un "gruppo WhatsApp" servendosi della funzione di "invio della posizione" messa a disposizione dall'applicazione WhatsApp stessa.

1	A	B	C	D	E	F	G	H	I	J
	Data	Giorno Settimana	Nome e Cognome	Flag attività aziendale	Gettone Feriale	Gettone Festivo	Flag before today	Macchine	Note	Note Macchine
799	04/09/2023	Lunedì		Y			VERO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
800	05/09/2023	Martedì					VERO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
801	06/09/2023	Mercoledì					VERO	JEEP	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
802	06/09/2023	Mercoledì					VERO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
803	06/09/2023	Mercoledì					VERO	JEEP	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
804	07/09/2023	Giovedì					VERO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
805	07/09/2023	Giovedì					VERO	JEEP	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
806	08/09/2023	Venerdì		Y			VERO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
807	08/09/2023	Venerdì					VERO	JEEP	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
808	09/09/2023	Sabato			Y		FALSO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
809	09/09/2023	Sabato					FALSO	JEEP	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
810	10/09/2023	Domenica					FALSO	JEEP	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
811	10/09/2023	Domenica					FALSO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
812	11/09/2023	Lunedì					FALSO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
813	12/09/2023	Martedì					FALSO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
814	13/09/2023	Mercoledì					FALSO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
815	14/09/2023	Giovedì					FALSO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
816	15/09/2023	Venerdì		Y			FALSO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
817	15/09/2023	Venerdì					FALSO	JEEP	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
818	16/09/2023	Sabato			Y		FALSO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
819	16/09/2023	Sabato					FALSO	JEEP	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
820	17/09/2023	Domenica			Y		FALSO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
821	18/09/2023	Lunedì					FALSO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
822	19/09/2023	Martedì					FALSO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
823	20/09/2023	Mercoledì					FALSO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	
824	21/09/2023	Giovedì					FALSO	GIULIETTA	JEEP PER UN ALTRO COLLEGA, TEMPO NON DEFINITO	

Figura 4.1. Foglio excel prenotazioni auto.

Il requisito espresso dall'azienda Mediamente Consulting consisteva pertanto nella realizzazione di una applicazione web con le seguenti funzionalità:

- visualizzazione della disponibilità di auto aziendali nella zona richiesta mediante una funzionalità di geolocalizzazione;
- inserimento di una richiesta di prenotazione;

- costruzione di un workflow approvativo sulla procedura di prenotazione delle auto;
- possibilità di inserire note in fase di richiesta di prenotazione al fine di fornire maggiori dettagli sulla richiesta e/o accordarsi con altri dipendenti.

### **4.0.3 Stakeholders**

Il progetto si rivolge alla totalità del personale dell'azienda Mediamente Consulting; analizzando gli stakeholders è stato possibile individuare due diverse categorie di attori che prendono parte al processo:

- il personale operativo dell'azienda (consulenti) che utilizza l'applicazione per prenotare un'auto, consultarne la disponibilità e tenerne aggiornata la posizione;
- i dirigenti aziendali che hanno la necessità di monitorare l'utilizzo del parco auto aziendale e di disporre di un processo approvativo semplificato.

### **4.0.4 Progettazione base dati**

Per lo sviluppo dell'applicazione web MMPrenotazioni Auto la base dati è rappresentata dal seguente schema UML.

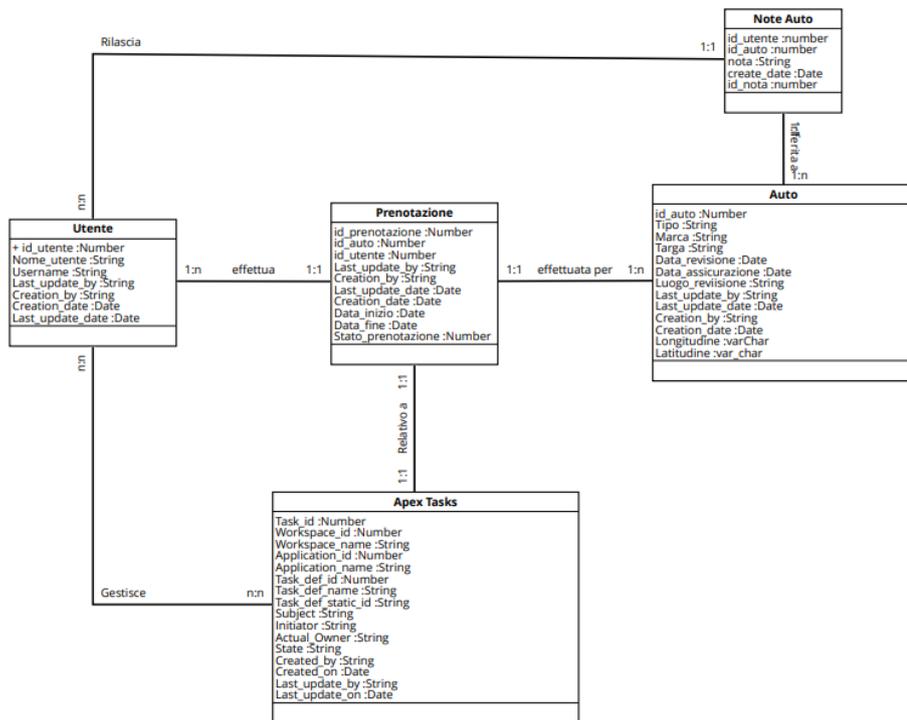


Figura 4.2. Base dati prenotazioni auto.

Sono state definite le tabelle:

- **Utente** per l'implementazione dell'anagrafica utente .
- **Prenotazione** destinata a contenere l'elenco delle richieste di prenotazione inserite dai vari utenti.
- **Auto** per l'implementazione dell'anagrafica auto contenente un set di informazioni aggiuntive sull'auto quali ad esempio la data scadenza revisione e la data scadenza assicurazione.
- **Note Auto** destinata a contenere eventuali commenti che un utente fa sulle auto per indicare ad esempio problematiche di natura meccanica riscontrate durante l'utilizzo dell'auto.

mentre la tabella Apex Tasks è una tabella che viene definita da Oracle Apex nel momento in cui si fa ricorso alla funzionalità di definizione/gestione dei task come verrà meglio dettagliato in seguito.

Alle tabelle Utente, Prenotazione e Auto sono associati dei Trigger ovvero delle procedure eseguite in modo automatico dal database quando si verifica un determinato evento come un inserimento, una modifica o una cancellazione. Si tratta sostanzialmente di procedure PL/SQL che consentono di specificare determinati vincoli di integrità sul database.

Per avere un maggiore controllo sugli elementi delle tabelle Utente, Prenotazione e Auto e per la gestione di imprevisti o errori, queste tabelle contengono i campi

- **Creation by:** indica il nome utente relativo all'utente che ha creato l'elemento a cui si riferisce.
- **Creation Date:** indica la data di creazione dell'elemento a cui si riferisce.
- **Last update by:** indica il nome utente che per ultimo ha apportato modifiche all'elemento.
- **Last update Date:** indica la data ultima in cui l'elemento è stato modificato.

sui quali agiscono i seguenti Trigger per governare in modo automatico le operazioni di inserimento e modifica dei record in tabella:

## Trigger Car\_prenotazione

```
create or replace trigger "BI_CAR_PRENOTAZIONE"
before insert on "CAR_PRENOTAZIONE"
for each row
begin
if :NEW."ID_PRENOTAZIONE" is null then
select "CAR_PRENOTAZIONE_SEQ".nextval into :NEW."ID_PRENOTAZIONE"
from sys.dual;
end if;
if inserting then
:new.creation_by := v('APP_USER');
:new.creation_date := sysdate;
:new.last_update_by := v('APP_USER');
:new.last_update_date := sysdate;
else
:new.last_update_by := v('APP_USER');
:new.last_update_date := sysdate;
end if;
end;
```

## Trigger Car\_ Utenti

```
        create or replace trigger "BI_CAR_UTENTI"
before insert on "CAR_UTENTI"
for each row
begin
    if :NEW."ID_UTENTE" is null then
        select "CAR_UTENTI_SEQ".nextval into :NEW."ID_UTENTE"
        from sys.dual;
    end if;

    if inserting then
        :new.creation_by := v('APP_USER');
        :new.creation_date := sysdate;
        :new.last_update_by := v('APP_USER');
        :new.last_update_date := sysdate;
    else
        :new.last_update_by := v('APP_USER');
        :new.last_update_date := sysdate;
    end if;

end;
```

## Trigger Auto

```
        create or replace trigger "BI_AUTO"
before insert on "AUTO"
for each row
begin
    if :NEW."ID_AUTO" is null then
        select "AUTO_SEQ".nextval into :NEW."ID_AUTO"
        from sys.dual;
    end if;

    if inserting then
        :new.creation_by := v('APP_USER');
        :new.creation_date := sysdate;
        :new.last_update_by := v('APP_USER');
        :new.last_update_date := sysdate;
    else
        :new.last_update_by := v('APP_USER');
        :new.last_update_date := sysdate;
    end if;

end;
```

#### 4.0.5 Modello di funzionamento dell'applicazione

Quando un utente (consulente) accede all'app mediante inserimento delle proprie credenziali, si possono presentare tre scenari distinti ovvero:

- l'utente entra nell'applicazione per effettuare una nuova richiesta di prenotazione;
- l'utente ha effettuato la prenotazione, che nel frattempo gli è stata approvata, e deve ottenere le coordinate aggiornate dell'auto richiesta;
- il dipendente aziendale ha terminato le sue attività e rilascia l'auto fornendo le nuove coordinate del luogo in cui è stata parcheggiata.

L'implementazione dei tre scenari appena descritti viene resa possibile mediante l'inserimento, in fase di progettazione, di opportuni Branches nella sezione Processing di Page Designer di Apex relativa alla pagina di login. Un branch è una funzionalità di Oracle Apex che permette di gestire un flusso di navigazione personalizzato all'interno di una pagina ed è caratterizzato da:

- un nome preferibilmente descrittivo dello scenario che va a sviluppare;
- un comportamento che deve realizzare;
- delle server side conditions che rappresentano le condizioni secondo cui il branch verrà eseguito.

Nel caso in analisi sono stati definiti i tre branch "Go To Prenota", "Go To Aggiorna Posizione" e "Go To Auto" per rispondere ai tre scenari sopra descritti.



Figura 4.3. Branches dell'applicazione.

In particolare

- ***Go to Prenota:*** personalizza il flusso di navigazione all'interno dell'app per quegli utenti che devono effettuare una richiesta di prenotazione. La funzione di questo primo branch è quella di reindirizzare l'utente alla pagina relativa al calendario delle prenotazioni, pagina attraverso la quale può essere effettuata la suddetta richiesta. La condizione necessaria affinché il reindirizzamento alla pagina di prenotazione venga eseguito si verifica quando l'utente accede all'applicazione senza aver già inserito una prenotazione per il giorno in cui avviene l'accesso.

Al login l'utente viene quindi indirizzato alla pagina di calendario riportata in figura e il cui funzionamento dettagliato verrà descritto più avanti.



Figura 4.4. Vista pagina di calendario.

- Go To Auto:** personalizza il flusso di navigazione all'interno dell'app per quegli utenti che, avendo effettuato la prenotazione che è stata nel frattempo approvata, devono ottenere le coordinate dell'auto prenotata. La funzione di questo secondo branch è quella di reindirizzare l'utente alla pagina di visualizzazione delle coordinate delle auto e la condizione affinché questo reindirizzamento avvenga consiste nell'effettuare l'accesso avendo una prenotazione attiva ed approvata con inizio non oltre un'ora dal momento del login. Al login l'utente viene quindi indirizzato alla pagina di visualizzazione delle coordinate auto illustrata in figura e il cui funzionamento dettagliato verrà descritto più avanti.

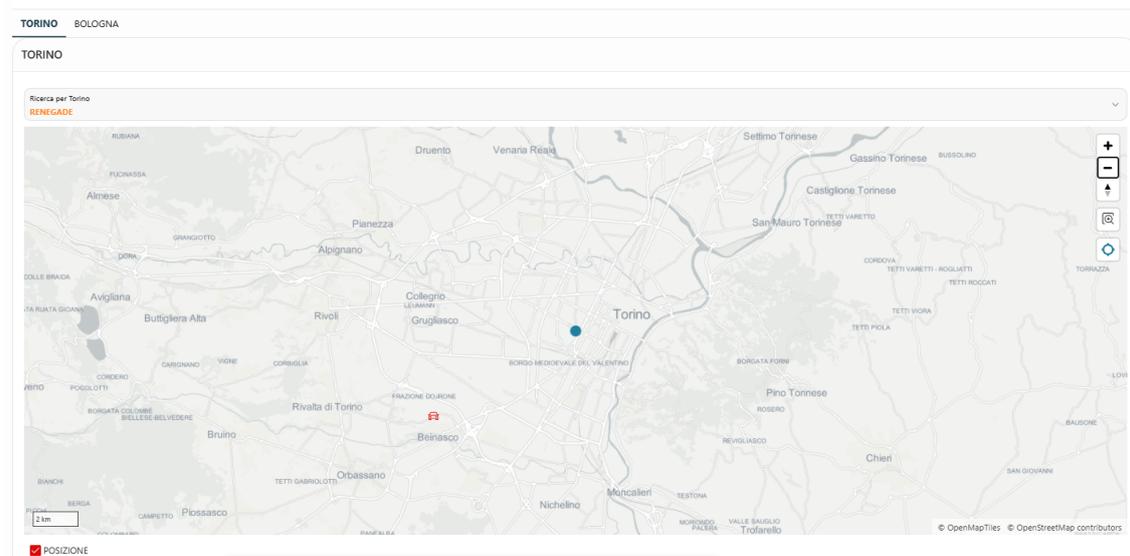


Figura 4.5. Mappa per raggiungere l'auto desiderata.

- **Go To Aggiorna Posizione:** personalizza il flusso di navigazione all'interno dell'app per quegli utenti che, dopo aver terminato le proprie attività presso la sede del cliente, rilasciano l'auto e devono caricare a sistema le nuove coordinate. La funzione di questo ultimo branch è quella di reindirizzare l'utente alla pagina di aggiornamento delle coordinate dell'auto; questo reindirizzamento verrà eseguito quando l'utente accederà all'app in un momento successivo alla scadenza del periodo per cui ha effettuato la prenotazione. Al login l'utente viene quindi indirizzato alla pagina di caricamento delle coordinate auto rappresentato in figura e il cui funzionamento dettagliato verrà descritto più avanti.

Carica posizione

Selezione Auto  
 RENEGADE

Latitudine  
 45.361641195416726

Longitudine  
 7.655469406287177

Error  
 Success

Aggiorna

Utente	Data Aggiornamento	Latitudine	Longitudine	Tipo Auto
LRITANO	14-OCT-2023 17:24	55.02548811301197	7.577093599666878	PUNTO
LRITANO	14-OCT-2023 17:23	45.02548811301197	7.577093599666878	PANDA
LRITANO	14-OCT-2023 17:23	45.02548811301197	7.577093599666878	PUNTO
LRITANO	14-OCT-2023 10:39	45.02547060129226	7.577071217710908	RENEGADE
LRITANO	13-OCT-2023 12:39	45.06166921321923	7.655473558308609	PANDA
LRITANO	13-OCT-2023 12:39	45.06166921321923	7.655473558308609	PUNTO
LRITANO	12-OCT-2023 21:55	45.0254713028872	7.577081567804561	PANDA
LRITANO	12-OCT-2023 21:55	45.0254713028872	7.577081567804561	RENEGADE
LRITANO	12-OCT-2023 12:03	45.061674418707085	7.655474198014597	RENEGADE
LRITANO	12-OCT-2023 11:54	45.06183557774966	7.655483536418172	PUNTO
LRITANO	12-OCT-2023 11:41	45.06165627368339	7.655474940366338	PUNTO
LRITANO	12-OCT-2023 11:39	45.061656169011556	7.65548340471242	PANDA

Figura 4.6. Pagina per l'aggiornamento coordinate.

Per ottenere questo comportamento dell'app, nella pagina di login, sono stati inseriti tre elementi di tipo "Branch" mediante la scheda Processing del relativo Page Designer. Di seguito vengono riportate le query eseguite per ciascuno dei tre scenari.

### Query branch scenario 1

```

select 1
from car_prenotazione p, car_utenti u
where u.id_utente=p.id_utente and
       u.username = :P9999_USERNAME and
       trunc(p.data_inizio,'DD') = trunc(sysdate,'DD') and
       p.stato_prenotazione=0
    
```

### Query branch scenario 2

```
select 1
from car_prenotazione p, car_utenti u
where u.id_utente=p.id_utente and
      u.username=:P9999_USERNAME
      and to_char(sysdate,'YYYYMMDDHH24MI') + 100
      > to_char(p.data_inizio,'YYYYMMDDHH24MI') and
      trunc(sysdate,'dd')= trunc(p.data_inizio,'dd') and
      p.stato_prenotazione=0;
```

### Query branch scenario 3

```
select 1
from car_prenotazione p, car_utenti u
where u.id_utente=p.id_utente and
      u.username=:P9999_USERNAME and
      to_char(sysdate,'YYYYMMDDHH24MI')- 100
      > to_char(p.data_fine,'YYYYMMDDHH24MI') and
      trunc(sysdate,'dd')= trunc(p.data_inizio,'dd') and
      p.stato_prenotazione=0;
```

## 4.1 Progettazione del sistema di prenotazione e relativo flusso approvativo.

L'analisi dei requisiti ed il confronto con gli attori aziendali coinvolti ha portato alla definizione del seguente modello di funzionamento del processo approvativo.

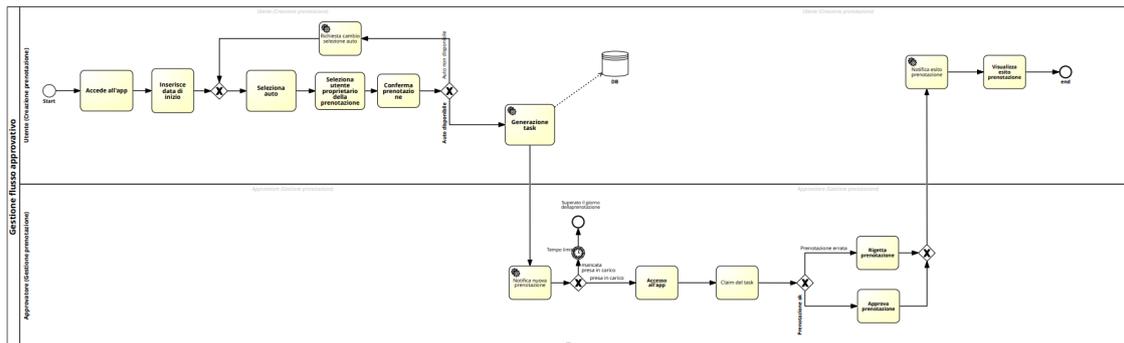


Figura 4.7. BPMN flusso approvativo.

### 4.1.1 Pagina nuova prenotazione

Chiariti i vari scenari e le conseguenti modalità di gestione verrà ora approfondito come l'utente si interfaccia alla pagina di prenotazione.

La pagina di prenotazione consiste sostanzialmente in un form nel quale l'utente dovrà inserire i seguenti dati:

- **Data inizio.**corrispondente a data e ora per cui si chiede la prenotazione di un'auto.
- **Data fine.**corrispondente a data e ora in cui si rilascerà l'auto avendone terminato l'utilizzo.
- **Nome Utente** per cui è richiesta la prenotazione.

- **Auto** che si intende prenotare.
- **Tipologia di attività** per cui si richiede la prenotazione dell'auto.

I controlli che vengono eseguiti sui dati in input consistono nell'assenza di sovrapposizione tra gli orari di prenotazione della stessa auto; due utenti, infatti, non possono richiedere la prenotazione di una determinata auto per lo stesso giorno e nello stesso orario.

Quando un utente seleziona un'auto per richiederne la prenotazione, se il coordinatore/approvatore ha già provveduto ad autorizzare una prenotazione per quell'auto nello stesso orario, il sistema notificherà la mancata disponibilità dell'auto all'utente che sta inserendo la richiesta di prenotazione. Invece, nel caso in cui la prenotazione antecedente non fosse già stata approvata spetterà all'utente coordinatore/approvatore approvare l'una e negare l'altra.

Le tipologie di controlli appena descritte in Oracle Apex vengono denominate Validations e per la loro implementazione occorre utilizzare la sezione Processi all'interno del Page Designer selezionando la voce Validations ed indicando il codice SQL che dovrà essere eseguito.

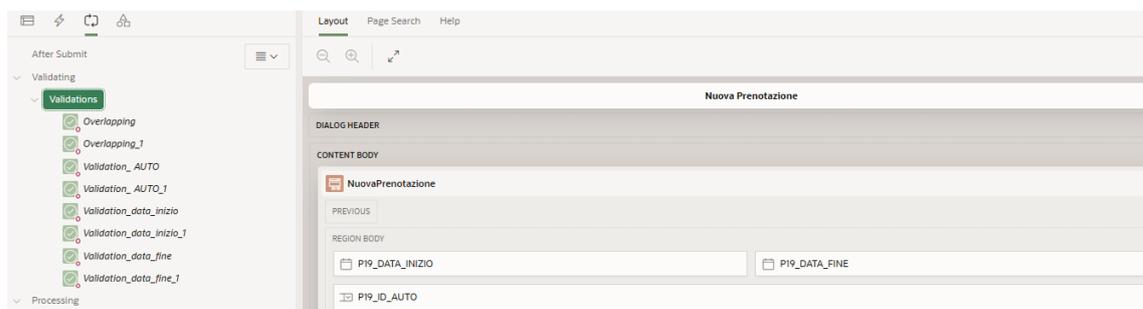


Figura 4.8. Validations nuova prenotazione

## Query validation

```
select 1
from CAR_prenotazione p
where p.data_inizio < TO_DATE(:P19_DATA_FINE, 'YYYY-MM-DD-HH24:MI:SS')
and
p.data_fine > TO_DATE(:P19_DATA_INIZIO, 'YYYY-MM-DD-HH24:MI:SS')
and
p.ID_AUTO = :P19_ID_AUTO
and
p.stato_prenotazione=0;
```

Qualora il suddetto controllo di Validation non venisse superato all'utente verrà notificata la mancata disponibilità dell'auto e potrà reinserire i dati selezionando un'auto differente.

Nel caso invece di controllo superato la prenotazione viene inserita nella tabella Car\_prenotazione e lo stato della prenotazione viene settato a "0", valore che indica che la richiesta di prenotazione è stata effettuata; la prenotazione inserita sarà visibile sul calendario in colore grigio chiaro e sarà in attesa di essere approvata dall'utente coordinatore/approvatore.

L'eliminazione di una prenotazione precedentemente inserita determinerà l'aggiornamento del campo STATO\_PRENOTAZIONE nella tabella Car\_prenotazione che verrà settato al valore '1'; in questo caso la prenotazione non verrà eliminata ma non risulterà più visibile sul calendario né comparirà come task da gestire all'interno del workflow di approvazione.

All'interno del calendario, per distinguere le prenotazioni ancora da approvare da quelle già approvate verranno utilizzati colori differenti: le prenotazioni ancora da approvare saranno visualizzate in grigio chiaro mentre le prenotazioni approvate

appariranno in colori diversi a seconda dell'auto prenotata.

Seguendo questa logica la query relativa agli elementi presenti in calendario sarà la seguente,

### Query calendario

```
SELECT
    P.ID_PRENOTAZIONE,
    A.TIPO,
    P.DATA_INIZIO,
    P.DATA_FINE,
    U.ID_UTENTE,
    U.GETTONI,
    P.stato,

    case when
    A.ufficio = 'TO' and p.tipologia_attivita=1
    then
    u.nome_utente||
    '(TO / '||A.TIPO||'
    - Attività privata - Gettoni utente' ||U.GETTONI||)')'
    when
    A.ufficio = 'TO' and p.tipologia_attivita=0
    then
    u.nome_utente||' (TO / '||A.TIPO||' - Attività Aziendale)'
    when
    A.ufficio = 'BO' and p.tipologia_attivita=0
    then u.nome_utente||' (BO / '||A.TIPO||' - Attività Aziendale)'
```

```
else
  u.nome_utente||
  '(BO / '||A.TIPO||'
  - Attività privata - Gettoni utente- ' ||U.GETTONI||')'
end TITOLO,
case
  when A.TIPO = 'PUNTO' then 'apex-cal-orange'
  when A.TIPO = 'PANDA' then 'apex-cal-red'
  when A.TIPO = 'SCENIC' then 'apex-cal-darkblue'
  when A.TIPO = 'RENEGADE' then 'apex-cal-green'
else 'apex-cal-gray'
end as css_class
FROM CAR_PRENOTAZIONE P, AUTO A, CAR_UTENTI U
WHERE P.ID_UTENTE= U.ID_UTENTE
      AND A.ID_AUTO=P.ID_AUTO
      AND U.ID_UTENTE=P.ID_UTENTE
      AND P.STATO_PRENOTAZIONE=0;
```

Parallelamente all’inserimento della prenotazione, nella tabella Car\_prenotazioni verrà creato un task per innescare il workflow del processo di approvazione come meglio dettagliato nel seguito.

#### 4.1.2 Definizione del Task di avvio workflow approvativo

Per implementare il requisito di costruzione di un workflow approvativo sulla procedura di prenotazione delle auto è stata utilizzata la funzionalità di Oracle Apex relativa ai Task.

In Apex un Task è l’oggetto sul quale lavorano le persone che partecipano ad un

workflow.

Gli elementi necessari per la definizione di un task sono: [30]

- **Partecipanti** cioè le tipologie di persone che possono partecipare all'esecuzione di un Task avendo permessi di azione diversi sull'attività alla quale si riferisce il task. Possono essere distinti in:
  - **Initiator:** è la persona che genera il Task.
  - **Potential Owner:** è la persona che dovrà prendere in carico il Task ed evaderlo (ad esempio approvando o negando una richiesta). È possibile definire, se serve, più di una persona che può prendere in carico un Task.
  - **Actual Owner:** è la persona che in un dato momento ha in carico un Task.
  - **Business Administrator:** è una persona con diritti amministrativi che può controllare in qualunque momento un Task e che ha i diritti amministrativi su tutti i Tasks creati.
- **Parametri:** sono gli attributi visibili all'interno di un task.
- **Azioni:** vale a dire tutti gli eventi che possono accadere nel ciclo di vita del task e che sono ricapitolabili nelle seguenti azioni:
  - **Claim:** permette ad un potential owner di prendere in carico il task (diventando così l'actual owner) riservandosi il diritto esclusivo di azione su di esso.
  - **Complete:** permette di svolgere l'azione necessaria al completamento del task
  - **Delegate:** permette ad un owner di delegare ad un potential owner la gestione del task rendendolo actual owner.

- **Update Comment:** permette di rilasciare commenti visibili da altri utenti sul task; ad esempio, un approvatore potrebbe inserire un commento per giustificare il rifiuto dell'approvazione.
- **Update Priority:** permettere di settare un livello di priorità differente al task.
- **Release:** permette all'actual owner di liberarsi del task rendendolo disponibile per un altro utente.
- **Cancel:** azione di "delete" del task utilizzata ad esempio da un utente che non ha più la necessità di effettuare una determinata richiesta di approvazione.
- **Create:** azione di creazione del task.

Per definire un task in Oracle APEX, occorre accedere alla sezione Shared Components dell'Application Builder, nella sezione Workflow and Authomation all'interno della quale sarà disponibile la voce Task definition. Come mostrato nelle immagini che seguono, per la definizione di un Task, Oracle Apex richiede di indicare alcuni parametri quali "Name", "Subject", "Static ID" e "Priority" e di collegare il task agli oggetti del database che sono impattati dalla sua esecuzione.

## MMPrenotazioni Auto

Task Definition: Richiesta prenotazione auto

Show All Name Settings Participants Parameters Actions Comments Last Updated

Name

Name: Richiesta prenotazione auto

Static ID: RICHIESTA\_PRENOTAZIONE\_AUTO

Settings

Subject: Prenotazione di &NAME\_UTENTE; Auto &TIPO; data inizio &DATA\_INIZIO; data fine &DATA\_FINE.

Priority: 3-Medium

Due On Interval: In 30 Minutes, In 1 Hour, In 6 Hours, In 12 Hours, Tomorrow, Next Week

Task Details Page URL: f?p=6APP\_ID,24:6SESSION::6DEBUG\_RP,24:P24\_TASK\_ID:6TASK\_ID.

Actions Source:  None  SQL Query  Table

Actions SQL Query

```
1 select p.id_prenotazione,
2 a.tipo,
3 u.nome_utente,
4 p.data_inizio,
5 p.data_fine,
```

Figura 4.9. Definizione task.

Task Definition: Richiesta prenotazione auto

Show All Name Settings Participants Parameters Actions Comments Last Updated

Participant Type Value Type Value

Participant Type	Value Type	Value
Potential Owner	Static	SMORASSUTTO

Parameters

Static ID	Label	Data Type	Required	Visible	Comment
data_fine	Data Fine	String	Yes	Yes	
data_inizio	Data Inizio	String	Yes	Yes	
nome_utente	Nome Utente	String	Yes	Yes	
stato	Stato	String	Yes	Yes	
tipo	Tipo	String	Yes	Yes	
Tipologia attivita	Tipologia Attivita	String	Yes	Yes	

Actions

Name	Outcome	Execution Sequence	Action Type
On Event: Complete			
SET_stato_prenotazione	Approved	10	Execute Code
delete_prenotazione	Rejected	20	Execute Code

Figura 4.10. Definizione task.

Il Subject che viene impostato per il Task indica le variabili che verranno valorizzate da Apex all'interno del Task. Nel caso specifico dell'applicazione MMPrenotazioni Auto il Subject del Task è stato definito sulle variabili

- *NOME\_UTENTE*
- *TIPO*
- *DATA\_INIZIO*
- *DATA\_FINE*

ed in questo caso, contestualmente all’inserimento di una richiesta di prenotazione, verrà generato un task che sarà valorizzato con in nome dell’utente che ha effettuato la richiesta, il tipo di auto richiesta, la data di inizio della prenotazione e la data di fine della prenotazione.

Una volta creato un Task (nel caso specifico una richiesta di prenotazione) questo dovrà essere preso in carico da una persona la quale sarà chiamata a svolgere una attività (ad esempio il responsabile dovrà approvare o rigettare la prenotazione). Le attività da svolgere possono essere definite direttamente all’interno del Task mediante delle Action (ad esempio Approve e Reject). Le Action di un Task servono per indicare cosa succede quando vengono effettuate delle operazioni sul task come, ad esempio, l’approvazione o il rigetto della richiesta di prenotazione mentre l’Action Source identifica all’interno di Apex i parametri che devono essere passati alle azioni del Task affinché questo possa eseguire le operazioni richieste. In questo caso i parametri per le Action del Task vengono passati mediante la query SQL:

## Query Task

```
select p.id_prenotazione,
       a.tipo,
       u.nome_utente,
       p.data_inizio,
       p.data_fine,
       p.tipologia_attivita
from car_prenotazione p, car_utenti u, auto a
where p.id_prenotazione=:APEX$TASK_PK and
       u.id_utente=p.id_utente and
       a.id_auto=p.id_auto;
```

Nell'applicazione MMPrenotazioni Auto all'interno del Task sono state create due Action rispettivamente per Approvare e per Rifiutare una richiesta. All'approvazione di una richiesta Apex dovrà aggiornare il Calendario impostando il colore definito per l'auto prenotata.

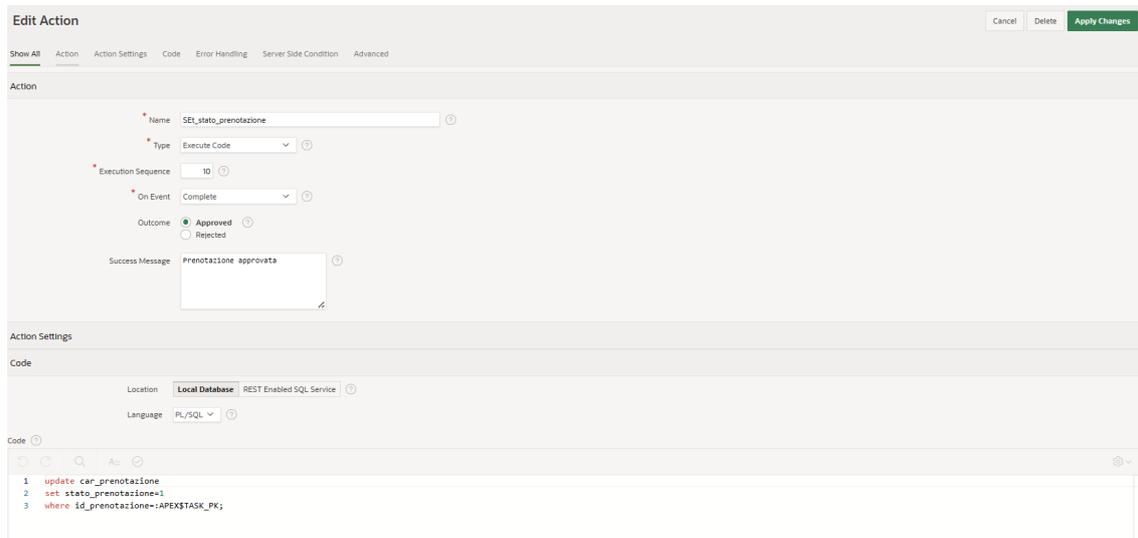


Figura 4.11. Azione eseguita all’approvazione del task.

Come mostrato nella figura sopra riportata, l’azione di approvazione sarà di tipo “esecuzione di codice PL/SQL” ed in questo caso Apex eseguirà la query seguente:

### Query On approve

```
update car_prenotazione
set stato_prenotazione=2
where id_prenotazione=:APEX$TASK_PK;
```

Il rifiuto di una richiesta comporterà invece come azione la cancellazione della richiesta dal calendario.

Anche in questo caso, l’azione sarà di tipo “esecuzione di codice PL/SQL” e la query eseguita sarà:

## Query On delete

```
update car_prenotazione
set stato_prenotazione=1
where id_prenotazione=:APEX$TASK_PK;
```

Per collegare il Task alla pagina di inserimento di una richiesta di prenotazione in Apex occorre creare un nuovo processo all'interno della pagina di prenotazione che scatterà al click sul bottone "Crea" nel form di inserimento richiesta e che inizierà il task con i parametri inseriti dall'utente come mostrato nella figura che segue

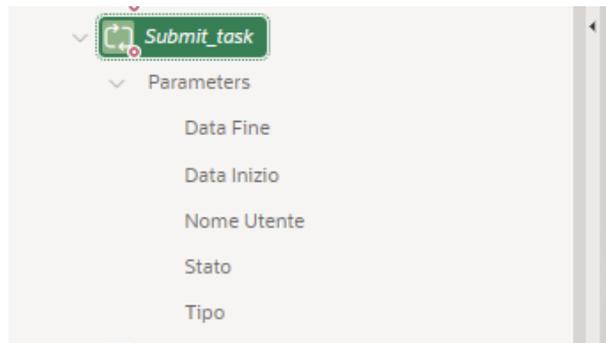


Figura 4.12. Processo di creazione task.

The screenshot shows the configuration page for a process in Oracle APEX. The process is named 'Submit\_task' and is of type 'Human Task - Create'. The configuration is organized into several sections:

- Identification:** Name: Submit\_task; Type: Human Task - Create; Editable Region: - Select -.
- Settings:** Definition: Richiesta prenotazione auto; Subject: (empty); Priority: (empty); Initiator Item: :APP\_USER; Task ID Item: (empty); Details Primary Key Item: P19\_ID\_PRENOTAZIONE.
- Execution Options:** Sequence: 20; Point: Processing; Run Process: Once Per Page Visit (default).
- Success Message:** (empty text area).
- Error:** (empty text area).

Figura 4.13. Dettagli processo creazione task.

Per comprendere il funzionamento dei Task e come questi possono essere utilizzati nell'implementazione di un workflow è opportuno comprendere il ciclo di vita di un task all'interno di Oracle Apex come spiegato nel paragrafo "Ciclo di vita di un task".

### 4.1.3 Unified task list

Per consentire agli utenti di gestire i tasks ai quali partecipano come attori all'interno di un workflow, Oracle Apex dispone di un componente ad hoc chiamato Unified Task List che consente di creare automaticamente delle pagine nelle quali saranno visibili tutti i Tasks ai quali un determinato utente partecipa. Le Unified Task List consentono di visualizzare un riepilogo dei tasks di un utente in una modalità simile a una casella di posta.

Mediante la creazione delle Unified Task List è possibile fornire a qualsiasi utente che partecipa ad un workflow una serie di elenchi di task con la distinzione tra:

- **Iniziati da me:** contiene l'elenco di tutti i task avviati dall'utente. Selezionando un task da questo elenco è possibile aggiornarne la priorità (se definita all'interno del progetto), annullare l'attività (ad esempio una prenotazione non più necessaria) o inviare informazioni ulteriori riguardanti il task (ad esempio le motivazioni per le quali si sta effettuando la prenotazione nel caso specifico dell'applicazione MMPrenotazioni Auto).
- **Le mie attività:** contiene l'elenco di tutti i tasks sui quali l'utente risulta come Actual Owner (ossia come utente assegnatario del task) e dei tasks nei quali l'utente risulta coinvolto come Potential Owner (ossia che dovrà ancora prendere parte attiva in qualche step del workflow). Sui task presenti all'interno di questo elenco si può ad esempio approvare o rifiutare l'attività (nello specifico approvare o rifiutare la prenotazione dell'auto) o richiedere ulteriori informazioni (ad esempio di fornire motivazioni valide per la richiesta di prenotazione).
- **Attività di amministrazione:** questa vista elenca tutti i tasks presenti nell'applicazione, inclusi quelli completati o annullati ed eventualmente quelli che presentano degli errori. L'amministratore può modificare la priorità dei

task (laddove sia stato definito un sistema di priorità). Nello specifico dell'applicazione MMPrenotazioni Auto sono state create tutte le tipologie di liste di task sopra elencate. In particolare un dipendente potrà vedere le sue prenotazioni nella lista Le mie richieste che sarà anche l'unica lista di task a lui visibile, un utente coordinatore/approvatore avrà invece a disposizione due liste: nella lista Approvazioni da effettuare vedrà l'elenco delle prenotazioni che risultano "assegnate" a lui per l'approvazione mentre nella lista Gestione richieste potrà vedere l'intero elenco delle prenotazioni che sono in attesa di approvazione da parte di uno qualsiasi degli utenti coordinatori/approvatori. Per una trattazione più dettagliata del processo di approvazione si rimanda al capitolo "Ciclo di vita di un Task".



Figura 4.14. Pagina Le mie richieste.

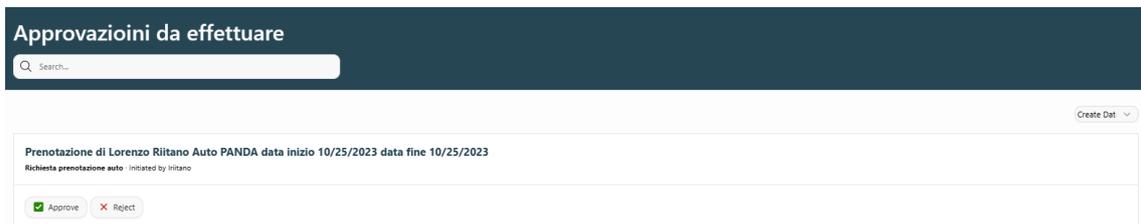


Figura 4.15. Pagina approvazioni da effettuare.

Nella pagina di definizione del Task riportata in figura 4.10 si può impostare la generazione automatica di una pagina, la Task Details Page, che Oracle APEX utilizzerà per consentire agli utenti di visualizzare i dettagli di un Task e che avrà il formato riportato nella figura seguente

**Task Details**

**Prenotazione di Lorenzo Riitano Auto PANDA data inizio 10/25/2023 data fine 10/25/2023**

Type	Richiesta prenotazione auto
Initiated	92 seconds ago
Initiator	Iriitano
Owner	smorassutto
Priority	Medium

Priority

**Details**

Data Fine	2023-10-25 23:59:00
Data Inizio	2023-10-25 00:00:00
Nome Utente	Lorenzo Riitano
Stato	0
Tipo	PANDA

**Comments**

Comment

Add Comment

**History**

Event	Time	User	Message
-------	------	------	---------

Figura 4.16. Pagina Task details.

#### 4.1.4 Ciclo di vita di un Task

Un task durante la sua vita può attraversare più stadi e trovarsi in stati differenti. Un task viene creato da un “Initiator”, si trova inizialmente in uno stato di “Unassigned” e compare nella lista dei task “da lavorare” di tutti gli attori che sono stati definiti come Potential Owner del task.

Nel caso specifico dell’applicazione MMPrenotazioni Auto il task viene generato dall’utente che effettua la richiesta di prenotazione di un’auto e comparirà tra i task da “lavorare” nella vista degli utenti coordinatori/approvatori. Un utente può essere impostato come coordinatore/approvatore settando a “Y” il campo Flag\_Admin presente nella tabella Utenti.

Come nota metodologica si può osservare che, qualora nell’elemento Task definito su Apex venisse individuato un unico “Potential Owner”, al momento della creazione di un task questo verrebbe immediatamente assegnato al suo Potential Owner e si troverebbe quindi subito in stato Assigned.

Nell’applicazione realizzata per Mediamente Consulting sono stati definiti più Potential Owner coincidenti con i responsabili aziendali e di conseguenza, all’inserimento di una richiesta di prenotazione da parte di un dipendente, tutti i responsabili (coordinatori/approvatori) vedranno nella propria pagina dei task da lavorare la richiesta appena inserita.

A questo punto un qualsiasi utente responsabile potrà prendere in carico l’attività facendo il “claim” del task e diventando così l’Actual Owner. Da questo momento il task si troverà nello stato “Assigned” e il coordinatore/approvatore potrà:

- Approvare la richiesta di prenotazione dell’auto.
- Rifiutare la richiesta di prenotazione dell’auto.
- Delegare un altro coordinatore/approvatore a lavorare la prenotazione.

Nel caso in cui la richiesta venga approvata il task passerà nello stato “Completed” e sul Calendario la prenotazione verrà colorata nel colore corrispondente all’auto

prenotata mediante l'esecuzione dell'Action "Approve" di cui si è parlato nel precedente punto 4.1.2.

Nel caso in cui la richiesta venga rifiutata il task passerà nello stato "Completed" e la prenotazione non risulterà più visibile sul Calendario mediante l'esecuzione dell'Action "Reject" di cui si è parlato nel precedente punto 4.1.2. Quando un utente responsabile delega un altro responsabile per l'approvazione il task rimane nello stato "Assigned" e come Actual Owner viene impostato il nuovo responsabile che troverà il task tra quelli a lui assegnati nella pagina personale dei task ma potrà in un secondo momento liberarsi del task compiendo un'azione di "release"; in questo caso il task ritornerà nello stato "Unassigned" e comparirà nuovamente nella lista dei task da lavorare di tutti gli utenti responsabili.

Oracle Apex consente anche di impostare un "Expiration Time" su un task; in questo caso, una volta trascorso l'expiration time da quando il task è stato creato e non ancora completato, il task sarà automaticamente settato come "Completed".

Un'ultima possibilità per un task consiste nell'essere cancellato dal suo initiator prima di essere completato da un potential owner, in questo caso il task verrà chiuso e chi ha preso in carico il task non dovrà più occuparsene.

Le figure che seguono riepilogano il "ciclo di vita" di un task all'interno di Oracle Apex e i possibili "diritti su un task" da parte delle varie tipologie di attori coinvolti in un workflow.

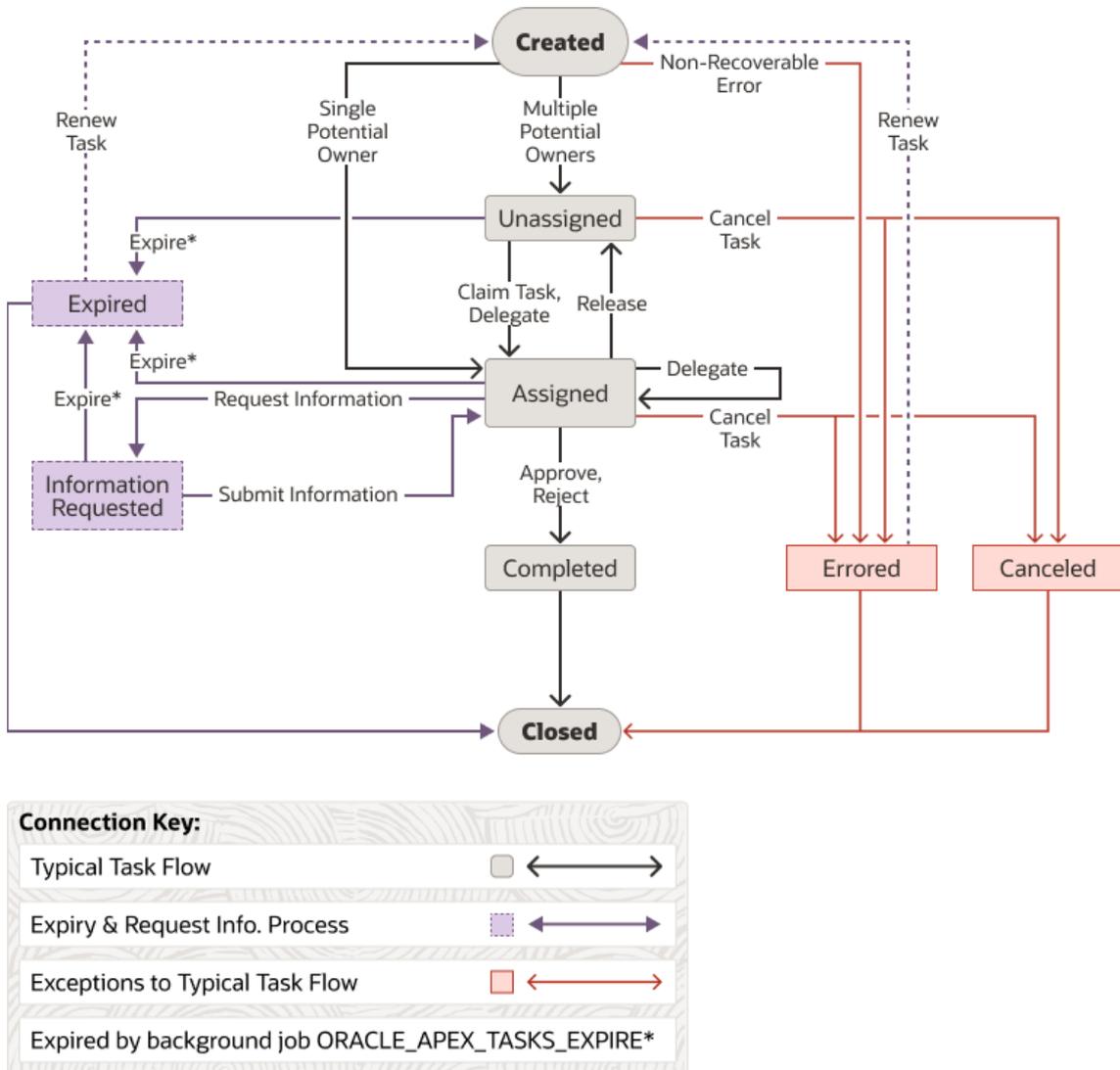


Figura 4.17. Ciclo vita di un task in Oracle Apex [12]

Operation	Description	Potential Owner	Actual Owner	Business Administrator	Initiator
<b>Claim</b>	Reserve a task exclusively to work on it.	✓	✗	✗	✗
<b>Approve</b>	Approve the task	✗	✓	✗	✗
<b>Reject</b>	Reject the task	✗	✓	✗	✗
<b>Release</b>	Release a previously claimed task so that it can be claimed by other potential owners	✗	✓	✗	✗
<b>Delegate</b>	Assign a task directly to a potential owner.	✗	✓	✓	✗
<b>Add Comment</b>	Add a comment to the task. E.g. Leave Request approver might leave a comment justifying why the request was rejected.	✓	✓	✓	✓
<b>Update Priority</b>	Change the priority of a task.	✗	✗	✓	✓
<b>Set Due Date</b>	Update the date on which this task is due	✗	✗	✓	✓
<b>Cancel</b>	Cancel the task. For e.g. an employee wanting to cancel his leave request because he no longer requires it.	✗	✗	✗	✓
<b>Add Potential Owner</b>	Adding one or more users to the list of potential owners who can act on the task.	✗	✗	✓	✗

Figura 4.18. Attori ed azioni di un task in Oracle Apex. [13]

#### 4.1.5 Progettazione Sistema di aggiornamento delle coordinate di un'auto

Una volta terminato l'utilizzo dell'auto aziendale il dipendente dovrà accedere all'app per aggiornare le coordinate dell'auto (vale a dire il luogo in cui l'auto è stata parcheggiata) secondo il modello BPMN riportato in figura.

#### 4.1 – Progettazione del sistema di prenotazione e relativo flusso approvativo.

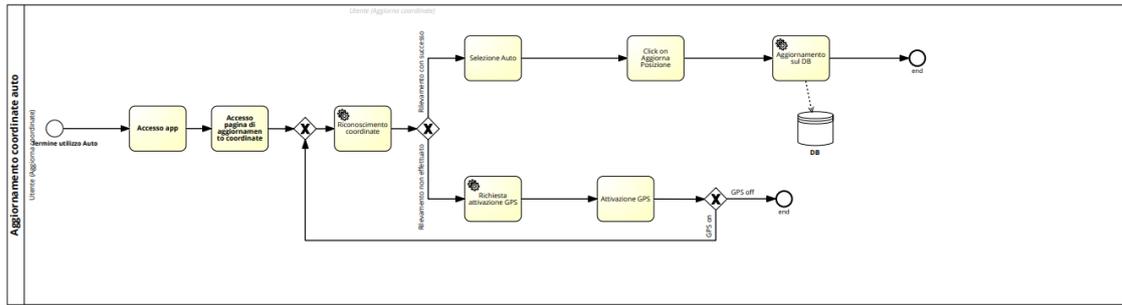


Figura 4.19. BPMN aggiornamento coordinate.

Per aggiornare le coordinate dell'auto l'utente eseguirà l'accesso all'app inserendo le proprie credenziali e verrà reindirizzato alla seguente pagina di aggiornamento.

**Carica posizione**

Selezione Auto: RENEGADE

Latitudine: 45.061641195416726

Longitudine: 7.655469406287177

Error: Success

Aggiorna

---

Search: All Text Columns Go Actions Reset

Utente	Data Aggiornamento	Latitudine	Longitudine	Tipo Auto
LRITANO	14-OCT-2023 17:24	55.02548811301197	7.577093599666878	PUNTO
LRITANO	14-OCT-2023 17:23	45.02548811301197	7.577093599666878	PANDA
LRITANO	14-OCT-2023 17:23	45.02548811301197	7.577093599666878	PUNTO
LRITANO	14-OCT-2023 10:39	45.02547060129226	7.577071217710908	RENEGADE
LRITANO	13-OCT-2023 12:39	45.06166921321923	7.655473558308609	PANDA
LRITANO	13-OCT-2023 12:39	45.06166921321923	7.655473558308609	PUNTO
LRITANO	12-OCT-2023 21:55	45.0254713028872	7.577081567804561	PANDA
LRITANO	12-OCT-2023 21:55	45.0254713028872	7.577081567804561	RENEGADE
LRITANO	12-OCT-2023 12:03	45.061674418707085	7.655474198014597	RENEGADE
LRITANO	12-OCT-2023 11:54	45.06163557774986	7.655483536418172	PUNTO
LRITANO	12-OCT-2023 11:41	45.06165627368339	7.655474940366338	PUNTO
LRITANO	12-OCT-2023 11:39	45.061656169011556	7.65548340471242	PANDA

Figura 4.20. Pagina aggiornamento coordinate.

Per consentire il caricamento delle coordinate all'interno della pagina di aggiornamento è stata utilizzata la funzionalità di Oracle Apex relativa alle Dynamics Actions; queste funzioni consentono allo sviluppatore di definire un comportamento lato client senza la necessità di utilizzare JavaScript per la definizione dell'evento ma limitando il suo impiego alla funzionalità necessaria.

Nel caso specifico, nella pagina “Carica posizione” presentata all'utente che entra nell'app per rilasciare un'auto appena utilizzata è stata creata una Dynamic Action che, all'apertura della pagina, eseguirà una funzione di geolocalizzazione e andrà a memorizzare longitudine e latitudine dell'utente nei due campi corrispondenti della pagina. A questo punto l'utente dovrà selezionare l'auto che sta rilasciando (l'auto da lui prenotata sarà preimpostata nella selezione ma potrà anche essere modificata) e cliccare sul pulsante “Aggiorna”. Questa azione comporterà la modifica nella tabella Auto del database delle informazioni su latitudine e longitudine della stessa.

## Funzione di geolocalizzazione

```
function getLocation() {
    var x=document.getElementById("P7_ERROR");
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(
            function setPosition(position) {
                //x.innerHTML="setPosition:"+position.coords.latitude;
                $s("P7_AGGIORNA_LATITUDINE" , position.coords.latitude);
                $s("P7_AGGIORNA_LONGITUDINE", position.coords.longitude);
            }
        function (error) {
            switch(error.code) {
                case error.PERMISSION_DENIED:
                    $s("P7_ERROR", "User denied the request for Geolocation.");
                    break;
                case error.POSITION_UNAVAILABLE:
                    $s("P7_ERROR", "Location information is unavailable.");
                    break;
                case error.TIMEOUT:
                    $s("P7_ERROR", "The request to get user location timed out.");
                    break;
                case error.UNKNOWN_ERROR:
                    $s("P7_ERROR", "An unknown error occurred.");
                    break;
            }
        }
        , {timeout:10000}
    );
}
```

```
        $s("P7_ERROR", "Success");
    }
    else
        x.innerHTML="Geolocation is not supported by this browser.";
}
```

La funzione 'getLocation()' gestisce la localizzazione dell'utente; viene eseguito innanzitutto un test per verificare se il software presente sul dispositivo dell'utente è idoneo alla rilevazione delle coordinate geografiche (GPS attivo e funzionante). Le informazioni relative a questo controllo vengono memorizzate nell'item P7\_ERROR della pagina Aggiorna posizione AUTO. Qualora la geolocalizzazione fosse avvenuta con successo vengono impostati i valori di latitudine e longitudine negli elementi con ID, P7\_LATITUDE , P7\_LONGITUDE presenti nella pagina di navigazione che rappresentano le coordinate geografiche della posizione utente. Per ottenere una risposta dal dispositivo dell'utente è impostato un valore di timeout pari a 10 secondi.

### 4.1.6 Procedura di accesso alle coordinate dell'auto

Come già detto, il branch “Go To Auto” personalizza il flusso di navigazione all'interno dell'app per quegli utenti che, avendo effettuato una prenotazione che è stata nel frattempo approvata, devono ottenere le coordinate dell'auto prenotata come mostrato dal BPMN in figura 4.21.

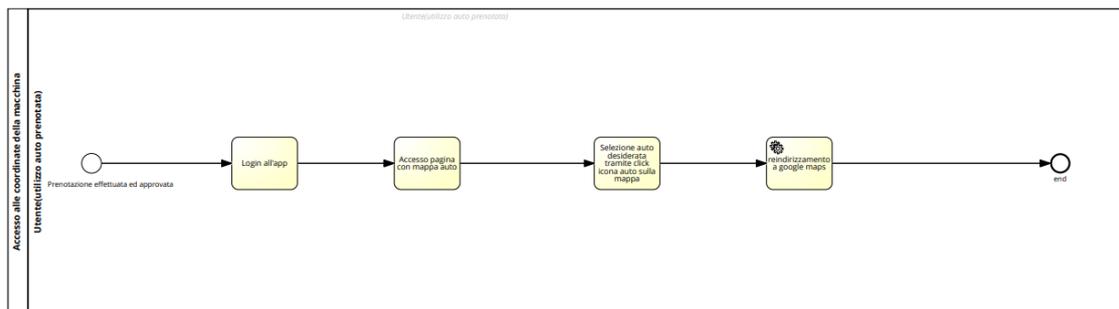


Figura 4.21. BPMN accesso auto.

Un utente che si trova a dover raggiungere l'auto prenotata verrà reindirizzato alla pagina denominata AUTO che presenta le seguenti tre sezioni:

- un'interactive grid contenente i dettagli di tutte le auto in dotazione all'azienda; si potrà consultare questa sezione per monitorare le scadenze di assicurazione, revisione e tagliando.

Un Interactive Grid è una tipologia di report che APEX consente di aggiungere ad una pagina dell'applicazione. Questa tipologia di report interattivo e personalizzabile presenta di default una barra di ricerca, un menù "Actions", un menù di intestazione delle colonne, e delle icone per l'update (Edit, Save e Add Row) di una riga. Il menu azioni consente agli utenti di modificare il layout del report selezionando ad esempio le sole colonne di interesse. Nello specifico dell'applicazione MMPrenotazioni Auto l'interactive grid consente di monitorare in tempo reale alcune informazioni sulle auto della flotta aziendale come la posizione (latitudine e longitudine) le scadenze di assicurazione, revisione e tagliando ed eventuali informazioni di servizio come l'indirizzo di riferimento per il "cambio gomme" o per eventuali "problematiche di manutenzione".

	Targa	Marca	Tipo	Ufficio	Longitudine	Latitudine	Data Revisione	Luogo Revisione	Tagliando	Km Al Tagliando	Autonoleggio	Scadenza Assicura	Indirizzo Manutaz	Indirizzo Cambio Go
<input checked="" type="checkbox"/>	FF222GG	FIAT	PUNTO	BO	7.57709359968...	55.0254881130...	9/11/2024					11/3/2023		
<input type="checkbox"/>	FK568TV	JEEP	RENEGADE	TO	7.57707121771...	45.0254706012...	11/16/2023					11/23/2023		
<input type="checkbox"/>	MM252TT	FIAT	PANDA	BO	7.57709359968...	45.0254881130...	11/16/2023					10/15/2023		
<input type="checkbox"/>	AS413LO	RENAULT	SCENIC	TO	7.65547496052...	45.0616503109...	12/13/2023					10/28/2023		

Figura 4.22. Interactive grid pagina Auto.

- una sezione contenente delle card; questa è una tipologia di template di reportistica che APEX mette a disposizione per visualizzare dati e informazioni raggruppate in schede o blocchi; le cards si presentano come schede colorate e visualizzano un set di informazioni estratte da una tabella sottostante. Una card può includere elementi interattivi (tipicamente delle Actions) che

consentono di effettuare delle azioni sugli elementi della tabella a cui fa riferimento.

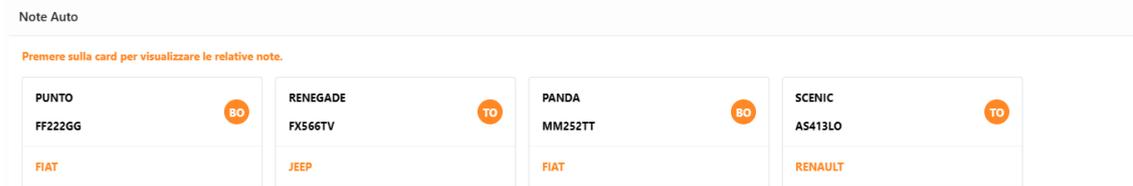


Figura 4.23. Cards per accedere alla chat relativa all'auto a cui si riferisce la card.

Nello specifico dell'applicazione MMPrenotazioni Auto, nella sezione cards vengono riportate sotto forma di icona tutte le auto della flotta aziendale; selezionando una card l'utente sarà reindirizzato ad una pagina contenente le note relative all'auto corrispondente. La pagina delle note consiste in una chat nella quale ogni utente può visualizzare i commenti sull'auto rilasciati dagli altri utilizzatori ed inserire eventualmente i propri commenti come mostrato in figura 4.24.

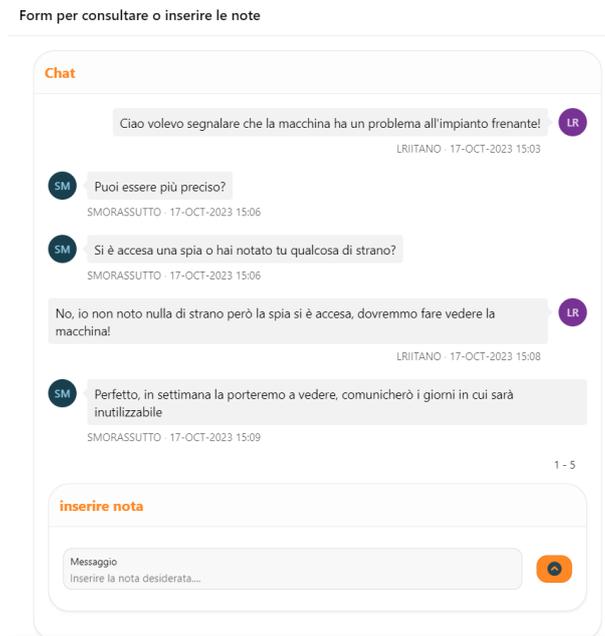


Figura 4.24. Regione per consultare/rilasciare commenti.

- una sezione che consente di visualizzare la posizione dell'auto all'interno di una mappa.

La sezione relativa alla mappa consente di visualizzare le auto, rappresentate da icone, all'interno di una mappa ed esegue il reindirizzamento a Google Maps per consentire all'utente di visualizzare il percorso da seguire per raggiungere l'auto prenotata.

Oracle APEX consente di implementare facilmente una mappa all'interno di una pagina selezionando semplicemente la voce "Map" come "Type" della regione corrispondente e definendone la fonte mediante una query SQL.

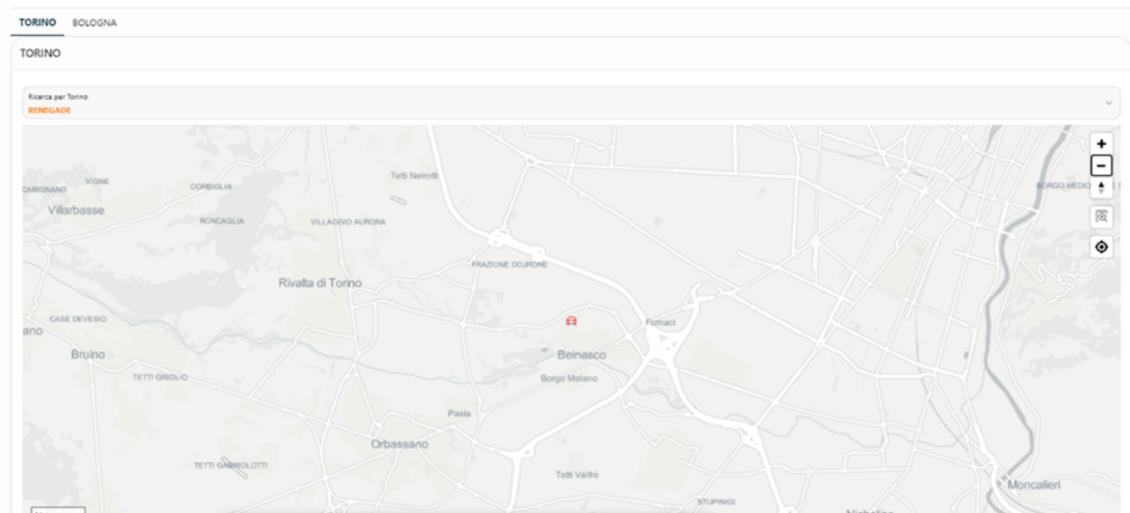


Figura 4.25. Regione per ottenere le coordinate dell'auto.

### Query definizione mappa

```
Select a.tipo,  
       a.LATITUDINE,  
       a.LONGITUDINE  
from AUTO a
```

Per consentire la visualizzazione del percorso da compiere per raggiungere l'auto prenotata, alla mappa definita nella pagina AUTO è stata associata una Dynamic action che ha la funzione di riconoscere la posizione attuale dell'utente, riconoscere le coordinate dell'auto prenotata e reindirizzare l'utente ad una pagina Google Maps che abbia come coordinate di partenza le coordinate dell'utente e come coordinate di arrivo le coordinate dell'auto.

La Dynamic action viene eseguita al click sull'icona raffigurante l'auto nella sezione

mappa della pagina AUTO; per implementare questo comportamento è stato necessario settare la Dynamic Action come “event Map Object Clicked”.

Al verificarsi dell’evento, ovvero del click sull’oggetto sono state impostate le seguenti Actions:

- una prima azione che esegue la stessa Funzione di geolocalizzazione già vista al precedente punto 4.1.5 per acquisire in questo caso la posizione dell’utente e settare i valori di latitudine e longitudine rispettivamente negli items P4\_LATITUDINE e P4\_LONGITUDINE;
- una seconda azione che esegue la funzione di reindirizzamento dell’utente alla pagina di Google Maps alla quale vengono passati i valori delle coordinate necessarie a tracciare il percorso tra la posizione dell’utente e la posizione dell’auto cercata.

### **Funzione sendGoogleMapsDir**

```
function sendGoogleMapsDir(lat_p, lng_p, lat_a, lng_a) {  
    window.open("https://www.google.com/maps/dir/"+lat_p+  
        "+lng_p+"/"+lat_a+", "+lng_a);  
}
```

Questa funzione permette di aprire google maps al click di un’icona sulla mappa, l’URL specifico che verrà aperto conterrà le coordinate dell’utente al momento del click sulla mappa e le coordinate dell’auto che sono state precedentemente aggiornate al termine di una precedente prenotazione.

## 4.2 **Sviluppi futuri**

In futuro verranno implementate funzionalità ancora da approfondire e rimandate ad un momento successivo alla prima diffusione dell'applicazione all'interno dell'azienda.

Una di queste riguarderà l'inserimento di un sistema di notifiche che avvisi i responsabili dell'avvenuto inserimento di una nuova richiesta da approvare e il dipendente nel momento in cui la sua richiesta di prenotazione è stata approvata.

Si potrebbe implementare, ad esempio, un sistema di notifiche push ovvero dei messaggi brevi consegnati direttamente sullo schermo dell'utente, sia esso mobile o desktop sullo stile dei messaggi WhatsApp o dei più classici sms. In alternativa alle notifiche push si potrebbe usare una notifica via e-mail nella quale, ad esempio, sia anche possibile inserire eventuali note che un responsabile ha aggiunto a motivazione del rifiuto di una prenotazione. Un altro sistema di notifica potrebbe essere rivolto ai gestori del parco auto aziendale per segnalare anomalie sulle auto riscontrate dai dipendenti durante il loro utilizzo nonché avvisi di scadenza della polizza assicurativa, del pagamento della tassa automobilistica o della revisione. Per quanto concerne il workflow approvativo, uno degli sviluppi futuri riguarderà l'implementazione di una funzionalità che renda possibile ad un utente l'inserimento di una prenotazione per un collega. Come sviluppo futuro di più ampio respiro l'applicazione potrà trasformarsi in un ambiente nel quale si possa tener traccia di tutte le trasferte effettuate dai dipendenti e che consenta di prenotare non solo le auto della flotta aziendale ma anche altri mezzi di trasporto come aerei e treni.

La gestione della totalità delle trasferte sarà resa possibile mediante l'implementazione all'interno dell'applicazione di un sistema di lettura dei codici QR e dall'integrazione di pagine che effettuino il reindirizzamento dell'utente ai siti di prenotazione di treni e aerei.



# Capitolo 5

## Conclusioni

Con riferimento all'esperienza di sviluppo descritta nel capitolo 4 si può confermare una discreta facilità d'uso della piattaforma Oracle APEX. Come in tutti i casi in cui ci si trova ad utilizzare un nuovo strumento, i primi approcci alla piattaforma non sono di immediata comprensione in quanto occorre acquisire la dovuta dimestichezza con il workspace e con i componenti presenti nell'Application Builder e nel Workshop. Il tutto per comprendere come gestire la creazione delle tabelle funzionali all'applicazione, per definire le pagine sulle quali dovranno agire gli utilizzatori e per impostare la navigazione all'interno dell'applicazione. Tuttavia, man mano che si approfondisce la conoscenza dello strumento, si comincia a percepire il risparmio di tempo offerto dalla piattaforma per la definizione dei vari oggetti dell'applicazione e per l'implementazione delle logiche di funzionamento delle varie pagine che si vanno a costruire.

A seguito dell'esperienza appena conclusa si può affermare che la piattaforma APEX non è di immediato e semplice utilizzo da parte di utenti/sviluppatori privi di un minimo di conoscenze di programmazione, di costruzione di database e soprattutto del linguaggio di interrogazione SQL, del linguaggio di programmazione procedurale PL/SQL e dei linguaggi HTML, CSS e JavaScript.

La piattaforma si connota pertanto come strumento ad utilizzo di sviluppatori mediamente esperti mentre si ritiene che non sia la soluzione ottimale per i cosiddetti citizen developer ai quali dovrebbero essere forniti strumenti “meno informatici”.

# Bibliografia

- [1] Creatio, “How much faster is low-code development comparing to traditional development?,” 2021. <https://www-statista-com.ezproxy.biblio.polito.it/statistics/1254662/low-code-development-speed-compared-traditional-it/>.
- [2] Mendix, “Usage of low-code for applications worldwide in 2021,” 2021. <https://www-statista-com.ezproxy.biblio.polito.it/statistics/1263327/low-code-benefits-organizations-use-applications/>.
- [3] AppMaster, “10 esempi di casi d’uso no-code e low-code,” 2022. <https://appmaster.io/it/blog/esempi-di-casi-duso-senza-codice-e-a-basso-codice>.
- [4] Gartner, “Low-code development platform market revenue worldwide from 2018 to 2025,” 2023. <https://www-statista-com.ezproxy.biblio.polito.it/statistics/1226179/low-code-development-platform-market-revenue-global/>.
- [5] S. Labs, “Low-code platforms impact on software quality worldwide in 2021,” 2023. <https://www-statista-com.ezproxy.biblio.polito.it/statistics/1226179/low-code-development-platform-market-revenue-global/>.
- [6] Mendix, “Gartner magic quadrant for enterprise low code application platforms 2022,” 2022. <https://www.mendix.com/resources/gartner-magic-quadrant-for-low-code-application-platforms/>.

- [7] Gartner, “Emerging technologies: The future of low code,” 2022.
- [8] ORACLE, “Oracle application express - deployment,” 2018. <https://www.oracle.com/database/technologies/appdev/apex/apex-deploy-installation.html>.
- [9] D. Peake, “Extending oracle e-business suite release 12 using oracle apex,” 2019. <https://www.oracle.com/technetwork/developer-tools/apex/learnmore/apex-ebs-extension-white-paper-345780.pdf>.
- [10] ORACLE, “Generating quick sql with sql shorthand syntax,” 2023. <https://docs.oracle.com/en/database/oracle/apex/23.1/aeutl/generating-quick-sql-with-sql-shorthand-syntax.html#GUID-A1308899-AA1D-42EA-8CAE-B128366538FE>.
- [11] R. Ahmed, *Oracle APEX 20 For Beginners: a platform to develop stunning, scalable data-centric web apps fast*. Independently Published, 2020.
- [12] Oracle, “About task states and transitions,” 2022. <https://docs.oracle.com/en/database/oracle/apex/22.1/htmdb/about-task-states-and-transitions.html#GUID-9A89E10C-0130-4C97-9197-F72546C36C89>.
- [13] A. Chatterjee, “Introducing approvals component in oracle apex,” 2022. <https://blogs.oracle.com/apex/post/introducing-approvals-component-in-oracle-apex>.
- [14] SAP, ““cos’è lo sviluppo di applicazioni low-code/no-code?”,” 2023. <https://www.sap.com/products/technology-platform/low-code/what-is-low-code-no-code.html>.
- [15] AppMaster, “Le migliori piattaforme low-code da utilizzare nel 2023,” 2023. <https://appmaster.io/it/blog/le-migliori-piattaforme-a-basso-codice>.
- [16] A. Ashok, “Low-code and no-code in 2021: Are they as useful as they seem?,” 2021. <https://www.forbes.com/sites/forbestechcouncil/2021/05/13/>

- low-code-and-no-code-in-2021-are-they-as-useful-as-they-seem/.
- [17] M. Griss, “Systematic software reuse: Architecture, process and organization are crucial,” 1996. <https://martin.griss.com/pubs/fusion1.htm>.
- [18] F. L. Trofa, “Sviluppo low code: cos’è e quali le migliori piattaforme,” 2023. <https://universeit.blog/sviluppo-low-code/>.
- [19] K. S.r.l., “Piattaforme di sviluppo low code e no code: vantaggi e casi d’uso,” 2023. <https://www.kinetikon.com/piattaforme-sviluppo-low-code-no-code-vantaggi-casi-uso/>.
- [20] D. Nazarevich, “Piattaforma low-code/no-code o sviluppo personalizzato? cosa scegliere?,” 2023. <https://innowise-group.com/it/blog/low-code-vs-no-code/>.
- [21] R. digital4, “Data-driven: cosa significa e perché un approccio basato sui dati è importante in azienda,” 2021. <https://www.digital4.biz/marketing/big-data-e-analytics/sei-regole-d-oro-per-un-data-driven-marketing-di-successo/>.
- [22] L. M. M. M. Antonio Corghi, Andrea Taglioni, “Il valore del dato,” 2022. <https://www.bip-group.com/wp-content/uploads/2022/05/Il-valore-del-dato.pdf>.
- [23] F. Torrini, “Approccio data-driven: come implementarlo efficacemente,” 2023. <https://universeit.blog/data-driven/>.
- [24] E. Covello, “Minimizzazione dei dati: consigli di accountability per diventare un’azienda data driven e sostenibile,” 2023. <https://www.cybersecurity360.it/legal/privacy-dati-personali/minimizzazione-dei-dati-consigli-di-accountability-per-diventare-unazienda-dat>
- [25] CoreSistemi, “Che cos’ è il quadrante magico di gartner?,” <https://coresistemi.it/quadrante-magico-di-gartner/>.
- [26] CoreSistemi, “Piattaforme low-code, i 5 fornitori leader per lo sviluppo delle business app,” 2023. <https://www.corrierecomunicazioni.it/tech-zone/>

- piattaforme-low-code-i-5-fornitori-leader-per-lo-sviluppo-delle-business-app/.
- [27] S. Lingua, “Oracle apex: piattaforma low-code come nuovo standard,” 2018. <https://blog.mediamenteconsulting.it/2021/03/18/mm039/>.
- [28] S. U. Foppa, “Il futuro del low code: verso strategie di business “componibili,”” 2023. <https://www.zerounoweb.it/cio-innovation/metodologie/il-futuro-del-low-code-verso-strategie-di-business-componibili/>.
- [29] M. Abbà, “Come mitigare le sfide di sicurezza del low-code/no-code,” 2023. <https://www.zerounoweb.it/techtarget/searchsecurity/come-mitigare-le-sfide-di-sicurezza-del-low-code-no-code/>.
- [30] Oracle, “Sql workshop guide disponibile release 23.1,” 2023. <https://docs.oracle.com/en/database/oracle/apex/23.1/aeut1/index.html#Oracle%C2%AE-APEX>.
- [31] F. L. Trofa, “Sviluppo low code: cos’è e quali le migliori piattaforme,” 2023. <https://universeit.blog/sviluppo-low-code/>.
- [32] Appian, “Il futuro dello sviluppo di applicazioni aziendali,” 2023. <https://www.globenewswire.com/en/news-release/2019/04/03/1795918/0/it/L-84-delle-organizzazioni-con-i-requisiti-pi%C3%B9-elevati-di-livello-enterprise-utilizza-lo-sviluppo-low-code-e-registra-ritorno.html>.
- [33] SlashData, “Leading low-code/no-code tools in 2022 worldwide, by developer usage,” 2022. <https://www-statista-com.ezproxy.biblio.polito.it/statistics/1343258/developer-usage-of-low-code-and-no-code-tools-worldwide/>.
- [34] AppMaster, “Come scegliere la giusta piattaforma low-code per la vostra azienda,” 2023. <https://appmaster.io/it/blog/come-scegliere-la-piattaforma-a-basso-codice>.