



4 Aprile 2025

# Yubiq Studio Training & Developer Manual

Rev. 5.1



---

WRITTEN BY:

---

VERIFIED BY:

---

APPROVED BY:

---

Avvale YUBIQ S.p.A.

Centro Direzionale Milanofiori – Strada 1 Palazzo F8, 20057 Assago (MI) – Italia – T + 39 02 57503366

C.F. e P.I. 11325150156 – Codice Destinatario 800HL4P – Registro Imprese di Milano n. 11325150156 – Numero REA MI-1458013

Capitale Sociale Euro 840.000,00 i.v. | [www.yubiq.com](http://www.yubiq.com)



# Document Change Log

Rev	Description	Date
5.0.14	Work in Progress	TBD

Avvale YUBIQ S.p.A.

Centro Direzionale Milanofiori – Strada 1 Palazzo F8, 20057 Assago (MI) – Italia – T + 39 02 57503366

C.F. e P.I. 11325150156 – Codice Destinatario 800HL4P – Registro Imprese di Milano n. 11325150156 – Numero REA MI-1458013

Capitale Sociale Euro 840.000,00 i.v. | [www.yubiq.com](http://www.yubiq.com)



# Index

<b>DOCUMENT CHANGE LOG .....</b>	<b>3</b>
<b>INDEX .....</b>	<b>4</b>
<b>INTRODUZIONE.....</b>	<b>6</b>
<b>LA METODOLOGIA ST.A.R. ....</b>	<b>7</b>
<b>Stages (Fasi).....</b>	<b>7</b>
<b>Actions (Azioni).....</b>	<b>7</b>
<b>Roles (Ruoli).....</b>	<b>7</b>
<b>WorkItem (Pratica) .....</b>	<b>7</b>
<b>Esempio di rappresentazione STAR .....</b>	<b>7</b>
<b>Yubiq Studio (YS).....</b>	<b>10</b>
<b>Installazione .....</b>	<b>11</b>
<b>La SandBox .....</b>	<b>12</b>
<b>TRAINING PROJECT: FLIGHT .....</b>	<b>13</b>
<b>Descrizione del processo.....</b>	<b>13</b>
I passi necessari.....	14
<b>Project e Workflow.....</b>	<b>15</b>
Modellazione di un Project .....	15
Modellazione di un Workflow.....	16
<b>Roles .....</b>	<b>17</b>
I ruoli Statici.....	17
I ruoli Dinamici.....	17
Modellazione dei Roles statici e dinamici.....	18
<b>SandBox Data: Users .....</b>	<b>19</b>
<b>Business Object.....</b>	<b>20</b>
Esempi di Business Object.....	20
Modellazione dei Business Object .....	21
Codifica dei Business Object.....	22

Avvale YUBIQ S.p.A.

Centro Direzionale Milanofiori – Strada 1 Palazzo F8, 20057 Assago (MI) – Italia – T + 39 02 57503366

C.F. e P.I. 11325150156 – Codice Destinatario 800HL4P – Registro Imprese di Milano n. 11325150156 – Numero REA MI-1458013

Capitale Sociale Euro 840.000,00 i.v. | [www.yubiq.com](http://www.yubiq.com)



<b>Business Object (EnumValue).....</b>	<b>24</b>
Modellazione dei Business Object (EnumValue) .....	24
<b>SandBox Data: Business Objects .....</b>	<b>25</b>
<b>Workitem.....</b>	<b>26</b>
Modellazione del WorkItem .....	26
Codifica c# del WorkItem .....	27
<b>Process Map.....</b>	<b>28</b>
Forms .....	29



# Introduzione

Questo documento ha l'obiettivo di integrare il manuale per lo sviluppo delle soluzioni BPM Yubiq con il relativo corso di formazione.

Dopo una panoramica sulla metodologia STAR, che è alla base della modellazione BPM, prenderemo un flusso di business di esempio, come la **richiesta di biglietti aerei**, e, implementandolo passo-passo, andremo a dettagliarne tutti gli aspetti legati allo sviluppo.

Yubiq Studio (da adesso YS) è uno strumento di sviluppo indirizzato a due categorie di specialisti:

## Analisti di business.



YS, con la sua modalità “Modeller”, permette la definizione delle strutture base senza scrivere codice (No-Code) come: **BusinessObjects**, **WorkItem**, **Stages**, **Actions**, **Forms**

## Sviluppatori C#.



La modalità “Developer” permette quindi di completare le restanti strutture come le **CodeLib** che vanno a contenere il codice macchina per le elaborazioni dati e le integrazioni oppure le **CustomViews** per creare oggetti grafici personalizzati.

Prima di addentrarci nello sviluppo con YS, introduciamo i concetti base della metodologia STAR. Questa non è una notazione o un formalismo come il BPMN, bensì una scomposizione di un processo di business in STages, Actions e Roles.



# La Metodologia ST.A.R.

La metodologia STAR (Stages, Action, and Roles) è stata sviluppata a Palo Alto, California, e rappresenta un approccio di Business Process Management (BPM) volto a migliorare l'efficienza e l'efficacia dei processi aziendali.

Rappresenta un approccio strutturato per implementare soluzioni di Business Process Management (BPM) che enfatizza l'importanza della chiarezza nelle fasi, nelle azioni da intraprendere e nei ruoli dei partecipanti. Ecco una panoramica dei principali elementi della metodologia:

## Stages (Fasi)

Le fasi rappresentano le diverse tappe del processo di gestione dei processi aziendali. Esempi validi possono essere:

- **Nuove Pratiche**
- **Pratiche da lavorare**
- **Pratiche in errore**
- **Archivio**

## Actions (Azioni)

Le azioni si riferiscono alle attività specifiche da compiere in ciascuna fase. Esempi validi possono essere:

- **Modifica i dati**
- **Avanza alla prossima Fase**
- **Assegna ad altro utente**

## Roles (Ruoli)

Definire chiaramente i ruoli è fondamentale per il successo della metodologia. Ogni partecipante ha responsabilità specifiche e il loro coinvolgimento è essenziale per un'implementazione efficace. I ruoli possono includere:

- **Richiedente**
- **Approvatore**
- **Ufficio Viaggi**
- **Direttore**

## WorkItem (Pratica)

Nella metodologia STAR, è presente anche il concetto di **WorkItem**, che rappresenta una specifica unità di lavoro assegnata a un individuo o a un gruppo all'interno di un processo. Un WorkItem può rappresentare qualsiasi pratica, attività o compito che deve essere completato affinché il processo possa continuare o raggiungere i suoi obiettivi.

## Esempio di rappresentazione STAR

Se prendiamo come esempio un progetto BPM per la “Richiesta di biglietti aerei” un esempio di **Stages** and **Actions** potrebbe essere il seguente:



1. **Inizio:**
  - Creazione della richiesta di volo da parte del richiedente.
  - Raccolta delle informazioni necessarie (dati del richiedente, dettagli del volo, ecc.).
2. **Verifica Preliminare:**
  - Controllo delle informazioni inserite.
  - Verifica dell'idoneità della richiesta (es. budget, policy aziendali, disponibilità).
3. **Approvazione:**
  - Inoltro della richiesta all'approvatore designato (supervisore o manager).
  - La richiesta è in stato di attesa di approvazione.
  - Il decisore può approvare, rifiutare o richiedere ulteriori informazioni.
4. **Ricerca Volo:**
  - Contattare i fornitori (agenzie di viaggio o sistemi di prenotazione) per ricerca opzioni di volo.
  - Raccolta delle opzioni di volo disponibili e relativi costi.
5. **Presentazione Opzioni:**
  - Presentazione delle opzioni di volo al richiedente.
  - Raccolta dei feedback o della scelta definitiva da parte del richiedente.
6. **Prenotazione:**
  - Conferma e prenotazione del volo selezionato.
  - Emissione dei biglietti e comunicazione delle informazioni al richiedente.
7. **Preparazione al Viaggio:**
  - Invio delle informazioni relative al viaggio (itinerario, registrazione, documentazione).
  - Verifica della conformità alle normative di viaggio (es. documentazione, bagagli, ecc.).
8. **Esecuzione del Viaggio:**
  - Il richiedente intraprende il viaggio.
  - Monitoraggio dell'andamento della richiesta (eventuali problemi o cambiamenti).
9. **Chiusura:**
  - Ritorno del viaggiatore.
  - Valutazione dell'esperienza di viaggio (feedback del richiedente).
  - Chiusura della richiesta e registrazione delle spese (rimborso, se applicabile).
10. **Analisi Post-viaggio:**
  - Raccogliere dati sulle spese effettive rispetto al budget.
  - Analisi delle performance del flusso di lavoro per future ottimizzazioni.

In una pratica di richiesta voli, un **WorkItem** può essere definito con diversi attributi per garantire una gestione efficace della richiesta. Ecco un elenco degli attributi più comuni che potrebbero essere associati a un WorkItem in questo contesto:

1. **ID del WorkItem:** Un identificatore unico per ogni richiesta di volo.
2. **Data di Creazione:** La data e l'ora in cui è stata generata la richiesta di volo.
3. **Richiedente:** Nome e dettagli di contatto della persona che fa la richiesta.
4. **Tipo di Voli:** Specifica se si tratta di un volo nazionale, internazionale, di andata e ritorno, ecc.
5. **Destinazione:** Luogo in cui si desidera volare.
6. **Data e Orario di Partenza:** La data e l'orario di partenza desiderati.
7. **Data e Orario di Ritorno** (se applicabile): La data e l'orario di ritorno desiderati.
8. **Numero di Passeggeri:** Quantità di persone che viaggeranno.
9. **Classe di Volo:** Classe di viaggio preferita (economica, business, first class).
10. **Motivo del Viaggio:** Descrizione del motivo per cui viene effettuata la richiesta (es. affari, vacanza, urgenza).
11. **Stato della Richiesta:** Lo stato attuale del WorkItem (ad esempio: "In attesa di approvazione", "Approvato", "Rifiutato", "Completato").

- 
12. **Data di Scadenza:** Un termine entro il quale la richiesta deve essere elaborata.
  13. **Note Aggiuntive:** Spazio per commenti o informazioni pertinenti inviate dal richiedente.
  14. **Assegnatario:** Persona o team responsabile della gestione della richiesta di volo.
  15. **Storico delle Modifiche:** Registro delle modifiche apportate alla richiesta (cambi di stato, aggiornamenti, ecc.).

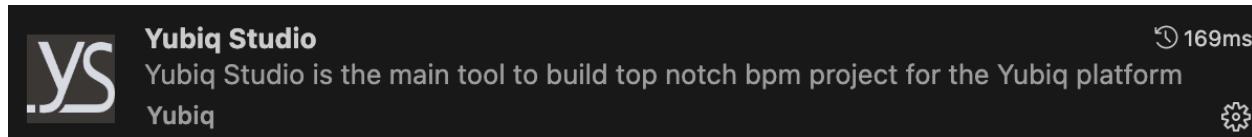
In conclusione, la metodologia STAR rappresenta un approccio innovativo e efficace per la gestione dei processi aziendali. Attraverso la chiara definizione delle fasi, delle azioni e dei ruoli, è possibile ottimizzare il flusso di lavoro, migliorare l'efficienza operativa e garantire una maggiore trasparenza e responsabilità all'interno dell'organizzazione. La sua applicazione in progetti BPM come la prenotazione di sale riunioni o la richiesta di biglietti aerei dimostra la versatilità e l'utilità di questo metodo. Con Yubiq Studio e la metodologia STAR, le aziende possono semplificare e automatizzare i loro processi, facilitando la gestione delle attività quotidiane e promuovendo una cultura di miglioramento continuo. L'integrazione di questi strumenti e metodologie offre un vantaggio competitivo significativo, contribuendo al successo e alla crescita sostenibile dell'organizzazione.



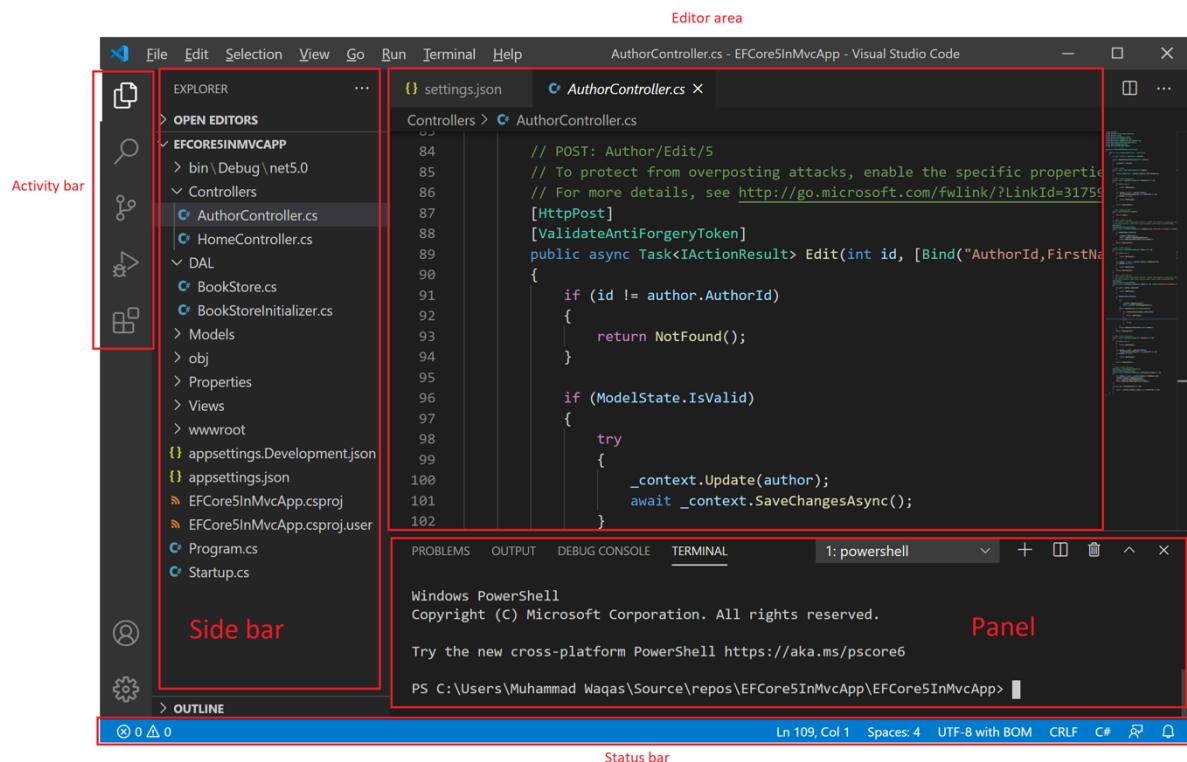
## Yubiq Studio (YS)

Yubiq Studio è basato interamente su **Visual Studio Code**, (VSC o vscode), è un editor di codice sorgente gratuito sviluppato da Microsoft per **Windows, macOS e Linux**. Le caratteristiche includono il debug, l'evidenziazione della sintassi, il completamento intelligente del codice, gli snippet, il refactoring del codice e il controllo di versione incorporato con Git. Con quasi il **75% della quota di mercato e 14 milioni di utenti**, VSCode è l'IDE più popolare disponibile (fonte Wikipedia).

Poiché VSC è un IDE generico ci sono oltre 30.000 estensioni nel suo store che consentono di personalizzare praticamente tutto nell'IDE. YS è formalmente una estensione di Visual Studio Code, scaricabile ed installabile dallo store Microsoft.



Yubiq Studio è uno strumento di sviluppo versatile e avanzato progettato per ottimizzare i processi aziendali e migliorare l'efficienza operativa. La sua interfaccia user-friendly e intuitiva consente agli sviluppatori di creare, gestire e automatizzare i WorkFlow senza dover ricorrere a soluzioni o script esterni. Grazie ai suoi Stages, Actions, Roles e WorkItem perfettamente strutturati e relazionati tra loro, Yubiq Studio offre un'esperienza di sviluppo integrata e coerente.



Durante la formazione con Yubiq Studio, gli utenti possono esplorare e utilizzare un unico "albero funzionale" che facilita la navigazione e l'interazione con gli oggetti del Project. Le Form dei WorkFlow sono completamente realizzabili con YS, utilizzando sia gli oggetti grafici standard, detti CoreView, sia il meccanismo delle CustomView per creare oggetti grafici personalizzati. Questo rende Yubiq Studio uno strumento potente e flessibile, adatto a soddisfare le esigenze specifiche di ogni organizzazione.



Inoltre, YubiQ Studio supporta l'integrazione con estensioni come Microsoft .NET C# e GitHub Copilot, che aggiungono ulteriori funzionalità di sviluppo e assistenza AI, rendendo il processo di sviluppo ancora più efficiente e produttivo.

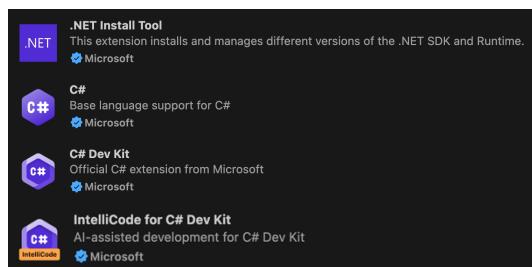
Con YubiQ Studio, le aziende possono semplificare e automatizzare i loro processi, facilitando la gestione delle attività quotidiane e promuovendo una cultura di miglioramento continuo.

## Installazione

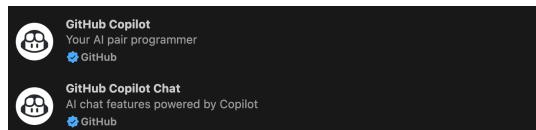
Una volta installato VSC su ambiente Windows o MAC (al momento Linux non è supportato) procedete con l'installazione dell'SDK .NET 9: <https://dotnet.microsoft.com/en-us/download/dotnet/9.0>

E quindi delle seguenti estensioni di VSC

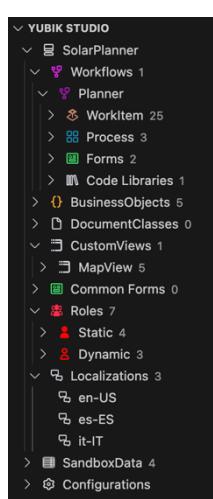
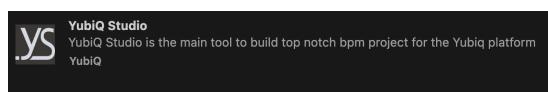
**Microsoft .NET C#:** Le estensioni base dello sviluppo .NET C#



**GitHub Copilot:** Per aggiungere l'AI durante lo sviluppo. Fortemente consigliato specie se disponete di una licenza d'uso piena!



**YubiQ Studio:** L'estensione per YubiQ



A questo punto VSC è pronto per la creazione di un nuovo progetto

Durante la formazione con YubiQ Studio, oltre agli **Stages**, **Actions**, **Roles** ed il **WorkItem** scoprirai altri oggetti perfettamente strutturati, relazionati tra loro ed esplorabili da un unico “albero funzionale”.

Tutto lo sviluppo di un **Project** viene realizzato esclusivamente con YS senza nessun ricorso a Solution o script esterni.

Tutte le **Form** di un **WorkFlow** si realizzano con YS e potranno essere utilizzati tutti gli oggetti grafici standard detti **CoreView** messi a disposizione. Se hai la necessità di realizzare un oggetto custom grafico ci viene in soccorso il meccanismo delle **CustomView** all'interno delle Form standard.

Avvale YUBIQ S.p.A.

Centro Direzionale Milanofiori – Strada 1 Palazzo F8, 20057 Assago (MI) – Italia – T + 39 02 57503366

C.F. e P.I. 11325150156 – Codice Destinatario 800HL4P – Registro Imprese di Milano n. 11325150156 – Numero REA MI-1458013

Capitale Sociale Euro 840.000,00 i.v. | [www.yubiq.com](http://www.yubiq.com)



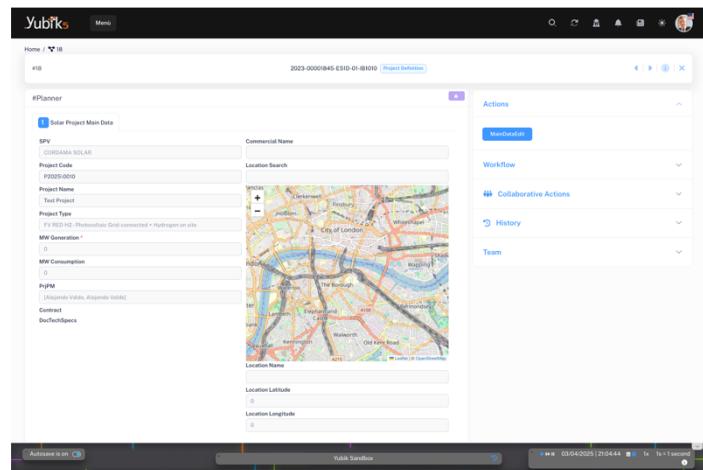
## La SandBox

La SandBox è un ambiente di sviluppo isolato che consente di simulare e testare un progetto senza influenzare il sistema di produzione reale. Questo strumento è essenziale per la fase iniziale dello sviluppo, poiché permette di lavorare su vari aspetti del progetto, come l'analisi, la modellazione dei flussi, la creazione delle strutture dati accessorie e la progettazione delle form di interazione. Inoltre, nella SandBox si può scrivere e testare tutto il codice che non dipende dall'ambiente di produzione YubiQ.

Utilizzando la SandBox, è possibile realizzare circa il 90% del progetto localmente, garantendo così una maggiore efficienza e velocità nella fase di sviluppo. Anche le fasi di dimostrazione del prototipo verso il cliente possono essere effettuate tramite la SandBox, che può essere resa "portatile" per facilitare le presentazioni in diversi contesti.

Un altro vantaggio dell'ambiente SandBox è la possibilità di utilizzare BusinessObject simulati per testare il funzionamento del progetto. Questo permette di verificare la corretta integrazione e interazione tra i vari componenti del sistema prima di passare alla fase di produzione.

La SandBox offre quindi un ambiente sicuro e controllato per l'ottimizzazione e la verifica del progetto.



YS permette di:

- Lanciare la SandBox localmente sul proprio Browser in qualsiasi momento dello sviluppo della soluzione. Ottimo per il costante test e Debug locale
- Esportare un file ZIP con la SandBox da lanciare su qualsiasi client Windows o Mac.
- Pubblicare la SandBox sul “SandBox Server” in modo che chiunque in possesso del link possa eseguirlo.

Avvale YUBIQ S.p.A.

Centro Direzionale Milanofiori – Strada 1 Palazzo F8, 20057 Assago (MI) – Italia – T + 39 02 57503366

C.F. e P.I. 11325150156 – Codice Destinatario 800HL4P – Registro Imprese di Milano n. 11325150156 – Numero REA MI-1458013

Capitale Sociale Euro 840.000,00 i.v. | [www.yubiq.com](http://www.yubiq.com)



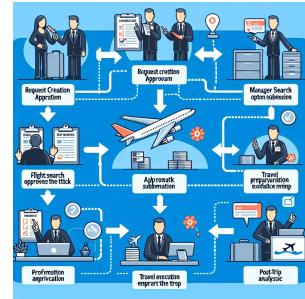
# Training Project: Flight

Il progetto Flight coordina il processo di richiesta dei biglietti aerei, includendo le varie approvazioni e l'emissione dei biglietti da parte dell'ufficio viaggi.

Prima di tutto, i dipendenti interessati devono presentare una richiesta con informazioni dettagliate riguardo alla destinazione, date e motivazioni del viaggio. Questa richiesta viene quindi sottoposta ai responsabili per l'approvazione, che valutano la necessità e la conformità delle spese alla politica aziendale. Una volta ottenuta l'approvazione, l'ufficio viaggi procede con la selezione delle opzioni di volo più convenienti e adatte alle esigenze temporali dei viaggiatori. Infine, dopo che il dipendente ha confermato la scelta, i biglietti vengono emessi e consegnati al richiedente, completando così il processo. Inoltre, l'ufficio viaggi fornisce assistenza continua per eventuali modifiche o imprevisti durante il viaggio.

## Descrizione del processo

- 1) Sia L'utente Dipendente che i Direttori hanno a disposizione la funzionalità per richiedere un biglietto aereo ed inserisce le informazioni necessarie.
- 2) Se è un Dipendente:
  - a) la sua richiesta è sottoposta all'approvazione del suo Manager.
  - b) Il Manager può fare delle domande di chiarimento al Dipendente che eventualmente può rinunciare al volo
  - c) Una volta ottenuta la approvazione la pratica passa all'ufficio viaggi
- 3) Se è un Direttore la pratica passa direttamente all'Ufficio Viaggi
- 4) L'ufficio Viaggi prenota il biglietto e passa alla pratica al richiedente
- 5) Il richiedente, entro tre giorni dalla data di partenza, può richiedere un cambio volo passando per una nuova approvazione del manager se dipendente o direttamente all'Ufficio Viaggi se Direttore.
- 6) Il richiedente, una volta ottenuto il biglietto può:
  - a) dare indicazione di aver effettuato il volo
  - b) Rinunciare al viaggio
  - c) Se passa più di una settimana dalla data di partenza il biglietto viene "rinunciato" automaticamente



Iniziamo a definire gli oggetti base di questo progetto. Ovvero i **Roles**, i **Business Object** ed il **WorkItem** inoltre dovremo creare i dati simulati per la **SandBox** per gli **Users** e i diversi **BusinessObjects (BO)**.



## I passi necessari

Per poter realizzare il processo descritto dovremo effettuare i passi illustrati nella tabella. Per ogni passo indichiamo la figura professionale più adatta tra il Modeller e il Developer.

Oggetto	Modeller	Developer
Project e Workflow	✓	
Roles	✓	
Business Objects	✓	
WorkItem	✓ Attributi base	✓ Attributi avanzati
SandBox Data	✓	✓
Process (Stages e Roles)	✓ Definizione	✓ Codice
Forms	✓ Definizione base	✓ Layout, Lookups, Validation
CodeLibs	✗	✓
CustomView	✗	✓

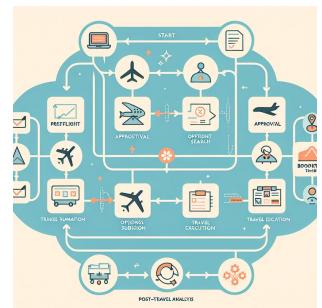
Nei prossimi capitoli andremo a dettagliare ognuno dei passi elencati facendo sempre attenzione a descrivere prima l'esempio pratico del progetto "Flight" e quindi gli aspetti realizzativi.



# Project e Workflow

La creazione di un Project è il primo passo da compiere per poter creare la struttura iniziale della soluzione. Una volta creato potremo andare a definire uno o più Workflows che ne comporranno la soluzione.

Creeremo quindi prima il Project **Flight** e quindi il Workflow **Flight**.



## Modellazione di un Project



### Step      Descrizione

1

Selezionare l'estensione YS sull'Activity Bar e quindi fare click su "New Project"

Verrà chiesto di indicare una cartella padre dove poter creare la nuova cartella del Project

Inserire quindi il nome del nuovo Project che chiameremo **Flight**

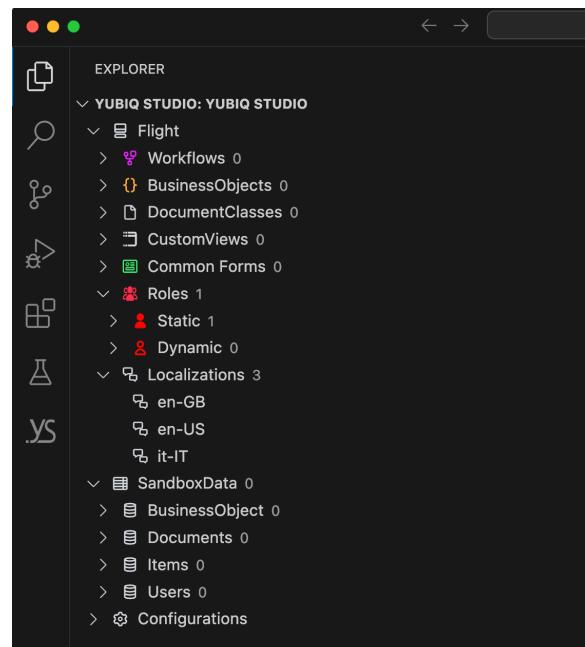
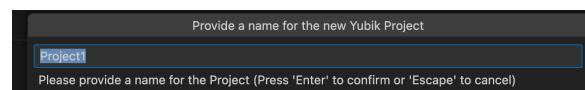
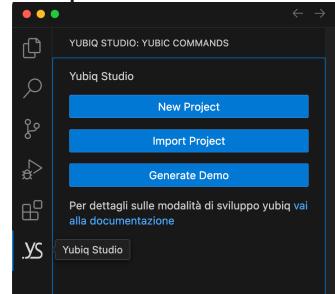
Contestualmente, sotto la cartella selezionata verrà creata una cartella con il medesimo nome

Individuate nella Side Bar l'albero YUBIQ STUDIO che dovrebbe presentarsi come in figura.

2 Uno dei primi rami è appunto il Workflows

Notate che non ha alcun Workflow definito

### Esempio





## Modellazione di un Workflow

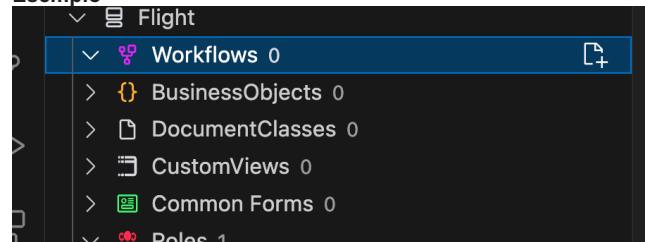


### Step      Descrizione

1

Individuate il ramo Workflows e fate click sull'icona

### Esempio

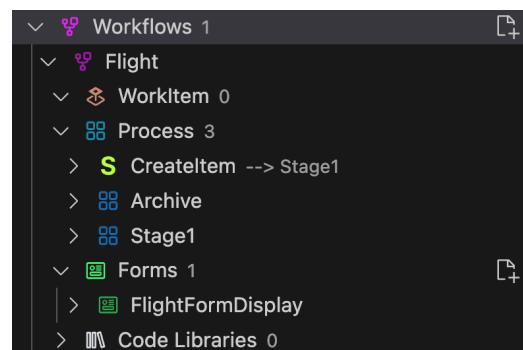


2

YS genererà la struttura iniziale del Workflow con:

- La Action di creazione
- Un primo Stage
- Lo stage finale di Archive
- Una Form minimale vuota

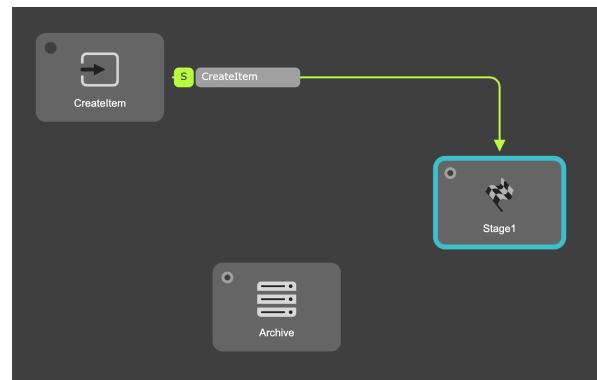
Questo aspetto permette l'esecuzione pressoché immediata della SandBox



4

Contestualmente con la generazione del codice necessario, viene anche generato il disegno del processo.

Facendo click sull'icona nel ramo del workflow appena creato questo viene mostrato





# Roles

Descriviamo i Ruoli che dovranno avere gli utenti distinguendoli tra quelli statici e quelli dinamici. Si ricorda che con ruolo statico indichiamo un gruppo di utenti fissi mentre un utente ottiene un ruolo dinamico basandosi sulle regole del processo.



## I ruoli Statici

I ruoli statici sono assegnati a un gruppo fisso di utenti all'interno di un sistema o di una soluzione. Questi ruoli non cambiano in base alle azioni o alle circostanze degli utenti, ma rimangono costanti nel tempo. Ad esempio, all'interno di una soluzione aziendale, potremmo avere due ruoli statici principali:

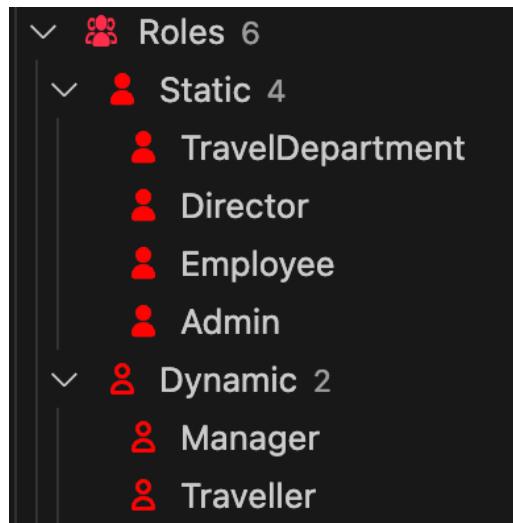
- **Employee:** Tutti i dipendenti dell'azienda appartengono a questo ruolo. Questi utenti hanno accesso a determinate risorse e funzionalità definite dalla loro posizione di dipendenti.
- **Director:** I direttori dell'azienda appartengono a questo ruolo. Hanno privilegi e accessi superiori rispetto agli Employee, in linea con le loro responsabilità direttive.

Inoltre, gli utenti dell'ufficio viaggi aziendale possono appartenere al ruolo statico TravelDepartment, che concede loro accessi specifici per la gestione dei viaggi aziendali.

## I ruoli Dinamici

I ruoli dinamici, al contrario, sono assegnati agli utenti in base a determinate condizioni o azioni all'interno del processo. Questi ruoli possono cambiare e non sono fissi, dipendendo dalle esigenze operative. Ad esempio:

- **Traveler:** Quando un utente effettua una nuova richiesta di biglietto aereo, acquisisce automaticamente il ruolo dinamico di Traveler. Questo ruolo consente all'utente di accedere a funzionalità specifiche relative alla gestione del viaggio.
- **Manager:** Quando un utente acquisisce il ruolo di Traveler, viene identificato il suo responsabile diretto, che acquisisce il ruolo dinamico di Manager. Per implementare questa logica, la definizione dell'utente viene estesa con l'attributo ManagedBy, che permette di identificare il responsabile.





## Modellazione dei Roles statici e dinamici



### Step      Descrizione

1 Posizionarsi sull'albero YS e selezionare il ramo "Roles". Compariranno i rami Static e Dynamic. Selezionate il sotto-ramo di interesse

Selezionate il sotto-ramo di interesse. Ad esempio Static

2 Fate quindi click sull'icona e nella barra in alto viene chiesto il nome da attribuirgli

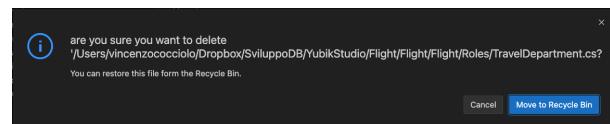
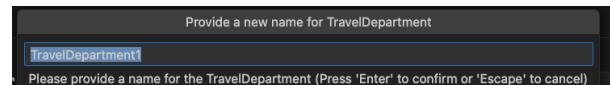
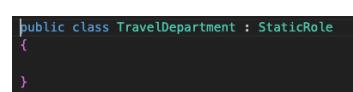
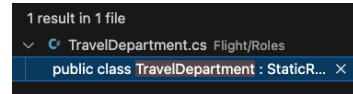
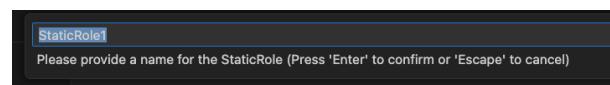
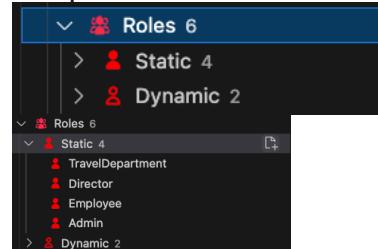
La procedura per i ruoli Dynamic è la medeima

Posizionandosi su ognuno dei singoli ruoli compaiono 4 diverse icone.

che rispettivamente permettono di

- 4
- Mostrare tutti i riferimenti del codice a quell'elemento
  - Mostra la parte di codice relativo alla definizione dell'elemento
  - Permette la modifica del nome
  - Cancellazione dell'elemento (previa conferma)

### Esempio





## SandBox Data: Users

Per poter eseguire in locale la SandBox e quindi poter provare il flusso è necessario creare i dati di esempio sotto forma di file Json.

Il primo file da editare è lo **users.json** il cui contenuto deve essere coerente con i Roles statici definiti e con gli attributi che un utente dovrà avere affinché la logica del workflow funzionai correttamente.



```
[  
  {  
    "Id": 1,  
    "UserName": "Vincenzo Cocciole",  
    "Roles": "Admin",  
    "ManagedBy": "",  
    "CompanyName": "Yubiq Italy SpA",  
    "Language": "it-IT",  
    "UserImg": "vc.png",  
    "Email": null  
  },  
  {  
    "Id": 2,  
    "UserName": "Davide Caricati",  
    "Roles": "Director",  
    "ManagedBy": "1",  
    "CompanyName": "Enel Energia SPA",  
    "Language": "it-IT",  
    "UserImg": "mam.png",  
    "Email": null  
  },  
]
```

Di fianco vedete l'esempio che viene creato automaticamente alla generazione del Project.

Quello che occorre fare è di editare la Key Roles inserendo tutti i ruoli statici separati da una virgola.

Per completare l'attività occorre inserire un ulteriore attributo:

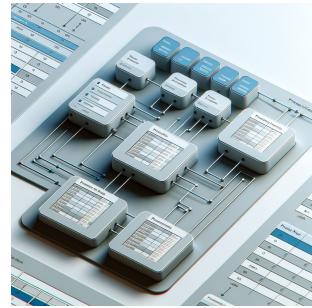
- **ManagedBy** con il nome dell'utente suo diretto responsabile



## Business Object

Come il nome suggerisce, i Business Object (BO) rappresentano entità del mondo reale, utilizzate nell'ambito della gestione dei dati e dei processi aziendali. Queste entità possono essere clienti, prodotti, ordini e molte altre figure essenziali per il funzionamento di un'azienda.

I Business Object sono modelli di dati che definiscono le caratteristiche e le proprietà delle entità aziendali. Ogni BO ha una serie di attributi che descrivono le sue proprietà e che possono variare a seconda della natura dell'entità stessa. Ad esempio, un BO Cliente potrebbe includere attributi come nome, indirizzo, numero di telefono e email.



### Esempi di Business Object

Cliente: Questo BO potrebbe avere attributi quali:

- ID Cliente (unico)
- Nome del cliente
- Indirizzo
- Numero di telefono
- Email

Prodotto: Questo BO potrebbe includere:

- ID Prodotto (unico)
- Nome del prodotto
- Categoria
- Prezzo
- Stock disponibile

Ordine: Gli attributi di questo BO potrebbero essere:

- ID Ordine (unico)
- ID Cliente
- ID Prodotto
- Data dell'ordine
- Quantità
- Stato dell'ordine

Per il workflow Flight Identifichiamo tre diversi BOs:

**Airline:** Elenco delle Linee aeree. Usato nelle form per la scelta della linea aerea. Il WorkItem ne referenzia una istanza

Airline		
ID	Int	1,2,3,4,5
Code	String	"AA", "BA", "DL"
Name	String	"American Airlines", "British Airlines", "Delta"

**Airport:** Elenco degli aeroporti. Usato nelle form per selezionare l'aeroporto di partenza e di ritorno. Il Workitem ne referenzia due istanze

Airport
---------

Avvale YUBIQ S.p.A.

Centro Direzionale Milanofiori – Strada 1 Palazzo F8, 20057 Assago (MI) – Italia – T + 39 02 57503366

C.F. e P.I. 11325150156 – Codice Destinatario 800HL4P – Registro Imprese di Milano n. 11325150156 – Numero REA MI-1458013

Capitale Sociale Euro 840.000,00 i.v. | [www.yubiq.com](http://www.yubiq.com)



ID	Int	1,2,3,4,5
Code	String	"CTA", "FCO", "LIN"
Name	String	"Catania Fontanarossa", "Roma Fiumicino", "Milano Linate"

**CostCenter:** Elenco dei centri di costo per l'attribuzione dei costi dei biglietti. Ogni utente appartiene ad un CostCenter. Il WorkItem ne referenza una istanza.

CostCenter		
ID	Int	1,2,3,4,5
Code	String	"HR", "MK", "FN"
Name	String	"Human Resources", "Marketing", "Finance"

## Modellazione dei Business Object



### Step      Descrizione

- 1 Posizionarsi sull'albero YS e selezionare il ramo "Business Objects".  
Fate quindi click sull'icona e nella barra in alto viene chiesto il nome da attribuirgli.

Inserire il nome **Airline**

- 2 Verrà creato in automatico il nuovo BO che include:
- Un elenco di attributo dove il principale è ID che ci dà l'univocità
  - Una o più Form che ci rappresentano il singolo BO

Posizionandosi sul nome del BO compaiono 4 diverse icone.

che rispettivamente permettono di

- 4
- Mostrare tutti i riferimenti del codice a quell'elemento
  - Inserisce un nuovo attributo del WorkItem
  - Permette la modifica del nome
  - Cancellazione dell'elemento (previa conferma)

Creiamo ora i restanti attributi del BO Airline. Ovvero **Name** e **Code** di tipo stringa.

- 5 Facendo click sull'icona compare l'elenco di tutti i possibili tipi di attributi che è possibile creare.  
Li vedremo in seguito quando parleremo del WorkItem. Limitatevi ora a scegliere il tipo "string"

- 6 Una volta completata l'attività il risultato dovrebbe essere come in figura.

### Esempio

Please provide a name for the BusinessObject (Press 'Enter' to confirm or 'Escape' to cancel)



Da notare che, se siete nella modalità Developer, compare al lato anche il codice C# auto generato.

Maneggiare con cautela!

```
public class Airline : BusinessObject
{
    [Key]
    2 references
    public virtual int Id { get; set; }
    0 references
    public virtual string Code { get; set; }
    0 references
    public virtual string Name { get; set; }
    0 references
    public override string GetId()
    {
        return Id.ToString();
    }

    0 references
    public override void SetId(string id)
    {
        Id = int.Parse(id);
    }
}
```

- 7 Completiamo l'attività creando anche i BO Airport e CostCenter

```
    < {} BusinessObjects 3
    > {} Airline 4
    > {} Airport 4
    > {} CostCenter 4
```

## Codifica c# dei Business Object

Una volta definito il Business Object si può passare alla fase di codifica C# per poter gestire aspetti più tecnici del loro funzionamento.

Al lato viene mostrato un BO appena generato. Nella prima riga identificate la decorazione DbBo questa sta ad indicare che il BO è di tipo database. YS al momento supporta solo questo tipo di BO mentre altri se ne aggiungeranno in seguito (Web Service, InMemory, Custom).

Gli attributi possibili di DbBo sono i seguenti

```
[DbBo]
public class Airline : BusinessObject
{
    [Key]
    public virtual int Id { get; set; }

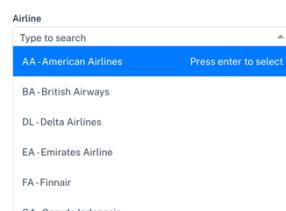
    public virtual string Code { get; set; }

    public virtual string Name { get; set; }
    public override string GetId()
    {
        return Id.ToString();
    }

    public override void SetId(string id)
    {
        Id = int.Parse(id);
    }
}
```

```
[DbBo(DescriptionProperties = new string[] { "Code", "Name" }, ConnectionName
="ExternalDB", TableName = "AIRLINE", IdPropName ="AL_ID")]
```

**DescriptionProperties:** è di tipo array di stringhe e indica i vari attributi che compongono la parte descrittiva dell'elemento. Questo viene tipicamente utilizzato negli elenchi a discesa delle Form, fornendo una descrizione chiara e sintetica degli elementi selezionabili.





**ConnectionName:** è di tipo stringa e specifica il nome della connessione al database utilizzata. Questo campo è fondamentale per garantire che il BO possa accedere correttamente ai dati memorizzati nel database una volta in produzione.

```
"dBConnections": [
    {
        "name": "YubikDB",
        "connectionString": ""
    },
    {
        "name": "ExternalDB",
        "connectionString": " Server=localhost;Database=ExternalDB;User Id=sa;Password=your_password;"
    }
],
```

**TableName:** è di tipo stringa e definisce il nome della tabella del database associata al BO. Attraverso questo attributo, il BO sa dove trovare i dati che deve manipolare.

**IdPropName:** è di tipo stringa e identifica il campo univoco della tabella del database. Questo campo è utilizzato per garantire l'unicità degli elementi all'interno della tabella, facilitando le operazioni di ricerca e gestione dei dati.

Al momento andremo solo a definire l'attributo “DescriptionProperties” con i campi “Code” e Name”



## Business Object (EnumValue)

Esiste anche una tipologia di BO più semplice costituito solo da un elenco di valori fissi come, ad esempio, la classe di un volo (Economy, Business, ...). Questa struttura è molto comoda da usarsi in quanto è collegata alle liste a discesa nelle Form.

```
[Flags]
public enum Class
{
    Economy = 1,
    Business = 2,
    First = 3,
    Premium = 4,
}
```

### Modellazione dei Business Object (EnumValue)



Step	Descrizione	Esempio
1	Posizionarsi sull'albero YS e selezionare il ramo "Business Objects" → "EnumValue"	

Fate quindi click sull'icona e nella barra in alto viene chiesto il nome da attribuirgli.

```
Class
Please provide a name for the Enum (Press 'Enter' to confirm or 'Escape' to cancel)
```

2	Inserire il nome <b>Class</b>
	Verrà creato in automatico il nuovo BO di tipo EnumValue vuoto

```
namespace Flight.BusinessObjects
{
    [Flags]
    2 references
    public enum Class
    {
    }
}
```

4	Inserite ora l'elenco dei valori testuali attribuendogli un numero identificativo come nell'esempio mostrato
---	--

```
[Flags]
2 references
public enum Class
{
    0 references
    Economy = 1,
    0 references
    Business = 2,
    0 references
    First = 3,
    0 references
    Magnifica = 4,
}
```

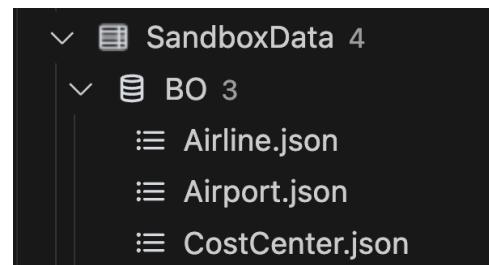


## SandBox Data: Business Objects

Una volta definiti i diversi BO della soluzione, è possibile creare all'interno della sezione SandboxData/BO i relativi file JSON che andranno a persistere i dati. Per fare ciò, si deve seguire un insieme di passaggi ben definiti:

Ognuno dei file JSON deve essere un array di oggetti. Questi oggetti devono presentare tutti gli attributi definiti nel BO. Questi attributi rappresentano le proprietà e le caratteristiche specifiche del BO, come ad esempio il nome, il tipo, e altri dettagli pertinenti.

```
[  
  {  
    "Id": "3",  
    "Code": "DL",  
    "Name": "Delta Airlines",  
    "BusinessObjectType": "Airline"  
  },  
  {  
    "Id": "4",  
    "Code": "EA",  
    "Name": "Emirates Airline",  
    "BusinessObjectType": "Airline"  
  },  
  {  
    "Id": "5",  
    "Code": "FA",  
    "Name": "Finnair",  
    "BusinessObjectType": "Airline"  
  },  
  {  
    "Id": "6",  
    "Code": "GA",  
    "Name": "Garuda Indonesia",  
    "BusinessObjectType": "Airline"  
  },  
  {  
    "Id": "7",  
    "Code": "HA",  
    "Name": "Hawaiian Airlines",  
    "BusinessObjectType": "Airline"  
  },  
]
```



Oltre agli attributi specifici del BO, è necessario aggiungere un attributo aggiuntivo chiamato "BusinessObjectType". Questo attributo avrà come valore il nome del BO relativo. Questo è fondamentale perché permette a YS di identificare facilmente il tipo di BO a cui appartiene ogni oggetto all'interno del file JSON.

I file JSON creati saranno utilizzati per la persistenza dei dati in tutte le prime fasi dello sviluppo senza dover accedere a risorse esterne come i Database server

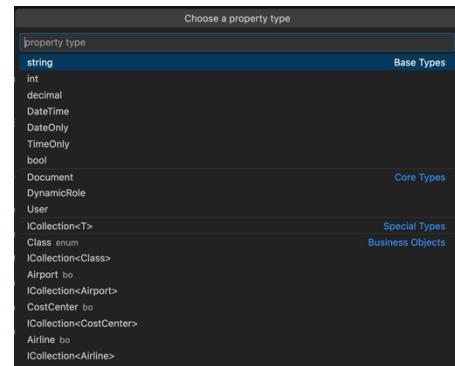


## Workitem

Il Workitem costituisce una delle strutture fondamentali di una soluzione, poiché include tutte le proprietà rilevanti relative alla pratica. Ogni Workitem rappresenta una istanza del processo che aggrega dati e documenti pertinenti, consentendo una gestione efficiente e integrata delle attività.

Un Workitem si compone di diverse tipologie di proprietà:

- **Base Types:** Campi standard singoli
  - String, Integer, Decimal, DateTime, Boolean
- **Core Typed:** Campi di strutture di sistema
  - Identificativo di un documento (Document)
  - Attributo di appoggio di un ruolo dinamico (DynamicRole)
  - Identificativo di un utente (User)
- **Business Objects:**
  - Singole istanze di uno specifico BO creato (Airline)
  - Insieme di istanze di uno specifico BO (iCollection<Airline>)



## Modellazione del WorkItem

### Step      Descrizione

1 Posizionarsi sull'albero YS e selezionare il ramo "WorkItem" del progetto Flight

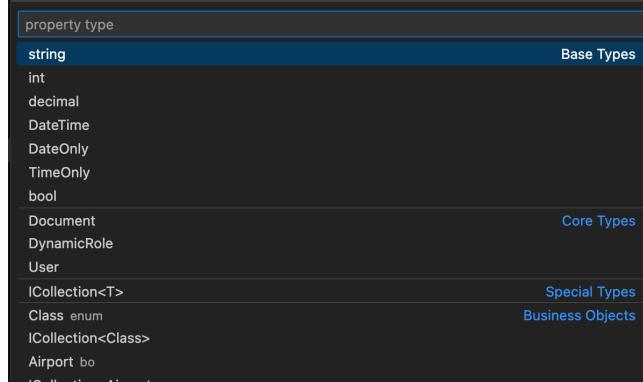
### Esempio



2 Fate quindi click sull'icona e nella barra in alto viene chiesto di selezionare il tipo di attributo da creare tra quelli elencati. Ad esempio string

Selezionato il tipo viene chiesto di definire il suo nome che deve essere univoco.

### Choose a property type

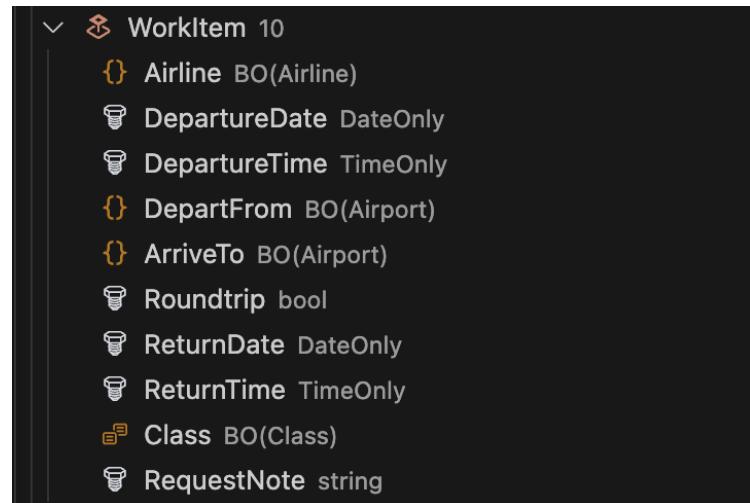


MyName

Please provide a name for the Property (Press 'Enter' to confirm or 'Escape' to cancel)



- 4 Ripetete l'operazione per tutte le proprietà da creare. Alla fine dovreste ottenere un elenco come in figura



## Codifica c# del WorkItem

Una volta definito il Business Object si può passare alla fase di codifica C# per poter gestire aspetti più tecnici del suo funzionamento.

In buona sostanza quello che viene richiesto è solo di specificare la dimensione in caratteri dei campi string.

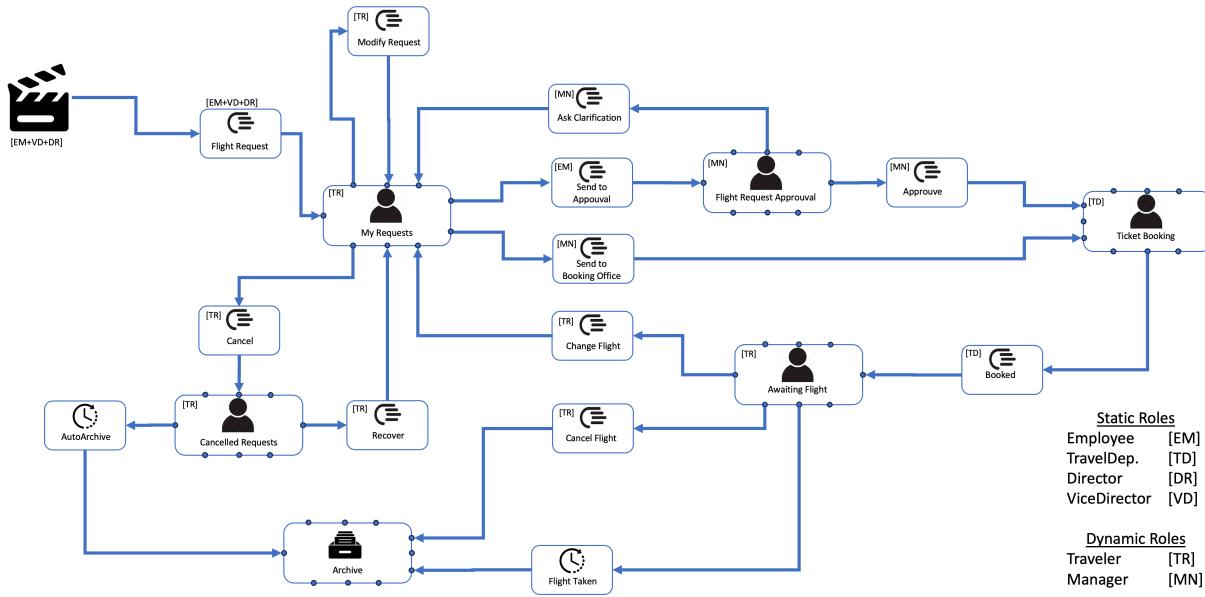
Nell'esempio al lato abbiamo impostato la string "RequestNote" a 1000 caratteri

```
public class FlightWI : WorkItem
{
    public virtual Airline Airline { get; set; }
    public virtual DateOnly DepartureDate { get; set; }
    public virtual TimeOnly DepartureTime { get; set; }
    public virtual Airport DepartFrom { get; set; }
    public virtual Airport ArriveTo { get; set; }
    public virtual bool Roundtrip { get; set; }
    public virtual DateOnly ReturnDate { get; set; }
    public virtual TimeOnly ReturnTime { get; set; }
    public virtual Class Class { get; set; }
    [MaxLength(1000)]
    public virtual string RequestNote { get; set; }
}
```



## Process Map

Di seguito viene rappresentata una possibile rappresentazione STAR del processo. Lo studente è invitato a modellare con YS lo stesso processo utilizzando gli stessi nomi degli Stage e delle Action.





## Forms

**Flight request:** Form di richiesta e modifica dei dati del biglietto richiesto. In modalità ReadOnly riporta le informazioni di richiesta. Dovrà richiedere la carta di identità del richiedente

Flight request	
Traveler	Cost center
<b>Outward journey</b>	
Airport	on 4/18/2024
	at 6:11:22
<b>Return journey</b>	
Airport	on 4/18/2024
	at 6:11:22
Preferred airline	Class
Reason	Cost £0.00
Details	
Raised by Metastorm	

**Flight reservation:** Form di inserimento e modifica dei dati del biglietto. In modalità ReadOnly riporta le informazioni del biglietto. Dovrà richiedere la foto del biglietto aereo

Flight reservation	
<b>Outward journey</b>	<b>Return journey</b>
Airport Metastorm	Airport Metastorm
on 4/18/2024	on 4/18/2024
at 6:12:32	at 6:12:32
Flight no. Metastorm	Flight no. Metastorm
eTicket no. Metastorm	

**VPAapproval:** Form di risposta nell'approvazione del VicePresident

**VP approval:** Form di risposta nell'approvazione del VicePresident

VP approval	
Special instructions?	
<input type="radio"/> Yes	<input type="radio"/> No



**Flightrequest:** Form di richiesta e modifica dei dati del biglietto richiesto. In modalità ReadOnly riporta le informazioni di richiesta. Dovrà richiedere la carta di identità del richiedente

**Flight request**

Traveler	Cost center
<b>Outward journey</b>	
Airport	on 4/18/2024
	at 6:11:22
Preferred airline	Class
Reason	Cost £0.00
Details	
Raised by Metastorm	

**Flight reservation**

<b>Outward journey</b>	<b>Return journey</b>
Airport Metastorm	Airport Metastorm
on 4/18/2024	on 4/18/2024
at 6:12:32	at 6:12:32
Flight no. Metastorm	Flight no. Metastorm
eTicket no. Metastorm	

**VPAapproval:** Form di risposta nell'approvazione del VicePresident

**FlightReservation:** Form di inserimento e modifica dei dati del biglietto. In modalità ReadOnly riporta le informazioni del biglietto. Dovrà richiedere la foto del biglietto aereo

**VP approval**

Special instructions?
<input type="radio"/> Yes <input type="radio"/> No
Comments