**Piano di Rifattorizzazione Graduale per PubliScript**

**1. Panoramica della Rifattorizzazione**

La rifattorizzazione di AIBookBuilder seguirà un approccio graduale per preservare la compatibilità con il codice esistente mentre si migliora l'architettura. L'obiettivo è separare le responsabilità mantenendo l'interfaccia pubblica per ridurre al minimo l'impatto sul codice client.

**Architettura Attuale vs. Target**

┌───────────────────────────────────┐ ┌───────────────────────────────────┐

│ ARCHITETTURA │ │ ARCHITETTURA │

│ ATTUALE │ │ TARGET │

├───────────────────────────────────┤ ├───────────────────────────────────┤

│ │ │ │

│ AIBookBuilder Monolitico │ │ PubliScriptApp (Facciata) │

│ │ │ │

│ ┌─────────┐ ┌────────┐ ┌───────┐ │ │ ┌─────────────────────────────┐ │

│ │Gestione │ │Gestione│ │Analisi│ │ │ │ AIBookBuilder │ │

│ │ UI │ │Browser │ │Mercato│ │ │ │(Mantenuto per compatibilità)│ │

│ └─────────┘ └────────┘ └───────┘ │ │ └─────────────────────────────┘ │

│ │ │ ↓ │

│ ┌─────────┐ ┌────────┐ ┌───────┐ │ │ ┌─────────────────────────────┐ │

│ │Gestione │ │Database│ │Genera-│ │ │ │ EventBus Centrale │ │

│ │ File │ │ │ │zione │ │ │ └─────────────────────────────┘ │

│ └─────────┘ └────────┘ └───────┘ │ │ ↓ │

│ │ │ ┌──────┬──────┬───────┬───────┐ │

│ ┌─────────────────────────────┐ │ │ │ UI │Browser│Storage│Analysis│ │

│ │ Altre Funzionalità │ │ │ │Manager│Control│Manager│Manager│ │

│ └─────────────────────────────┘ │ │ └──────┴──────┴───────┴───────┘ │

└───────────────────────────────────┘ └───────────────────────────────────┘

**2. Fasi di Rifattorizzazione**

**Fase 1: Preparazione e Riorganizzazione**

| **Obiettivo** | **Creare la base per la rifattorizzazione mentre si mantiene tutto funzionante** |
| --- | --- |
| Durata | 1-2 settimane |
| Priorità | Alta |

**Attività:**

1. Creare il pattern "Facade" per AIBookBuilder
2. Implementare l'EventBus basico
3. Creare StorageManager per gestione file/risultati
4. Sviluppare ResultsViewer indipendente

**Fase 2: Estrazione Browser Controller**

| **Obiettivo** | **Isolare la logica di interazione con il browser** |
| --- | --- |
| Durata | 1-2 settimane |
| Priorità | Alta (risolve i problemi più critici attuali) |

**Attività:**

1. Estrarre la logica del browser in BrowserController
2. Adattare AIBookBuilder per usare il nuovo controller
3. Migliorare la gestione degli errori del browser

**Fase 3: Riorganizzazione Analisi di Mercato**

| **Obiettivo** | **Specializzare la logica di analisi del mercato** |
| --- | --- |
| Durata | 2-3 settimane |
| Priorità | Media |

**Attività:**

1. Creare AnalysisManager specializzato
2. Rifattorizzare i metodi di analisi
3. Migliorare la persistenza dei risultati

**Fase 4: Gestione UI Specializzata**

| **Obiettivo** | **Isolare la gestione dell'interfaccia utente** |
| --- | --- |
| Durata | 2-3 settimane |
| Priorità | Media-Bassa |

**Attività:**

1. Creare UIManager dedicato
2. Migliorare l'aggiornamento dei componenti Gradio
3. Implementare sistema di feedback all'utente più robusto

**Fase 5: Generazione Libro e Completamento**

| **Obiettivo** | **Specializzare la generazione dei libri e completare la migrazione** |
| --- | --- |
| Durata | 2-3 settimane |
| Priorità | Bassa |

**Attività:**

1. Creare BookGenerator specializzato
2. Finalizzare la classe PubliScriptApp
3. Documentare la nuova architettura

**3. Mappatura Dettagliata dei Metodi**

**Tabella di Migrazione dei Metodi**

| **Metodo Attuale in AIBookBuilder** | **Classe Destinazione** | **Nuovo Nome Metodo** | **Fase** |
| --- | --- | --- | --- |
| **Metodi di UI** |  |  |  |
| create\_interface() | UIManager | create\_interface() | 4 |
| show\_feedback() | UIManager | display\_feedback() | 4 |
| update\_analysis\_status() | UIManager | update\_status() | 4 |
| debug\_check\_components() | UIManager | verify\_components() | 4 |
| **Metodi di Analisi** |  |  |  |
| analyze\_market() | AnalysisManager | run\_analysis() | 3 |
| \_analyze\_market\_crisp() | AnalysisManager | run\_crisp\_analysis() | 3 |
| \_analyze\_market\_legacy() | AnalysisManager | run\_legacy\_analysis() | 3 |
| \_filter\_legacy\_prompt\_sections() | AnalysisManager | filter\_prompt\_sections() | 3 |
| get\_selected\_phases() | AnalysisManager | get\_active\_phases() | 3 |
| **Metodi di Browser** |  |  |  |
| connect\_callback() | BrowserController | connect() | 2 |
| send\_to\_genspark() | BrowserController | send\_message() | 2 |
| get\_last\_response() | BrowserController | get\_response() | 2 |
| check\_for\_generation\_error() | BrowserController | check\_errors() | 2 |
| handle\_context\_limit() | BrowserController | manage\_context\_limit() | 2 |
| reset\_context\_manual() | BrowserController | reset\_context() | 2 |
| wait\_for\_stable\_response() | BrowserController | wait\_for\_response() | 2 |
| add\_url\_change\_monitor() | BrowserController | monitor\_url\_changes() | 2 |
| take\_debug\_screenshot() | BrowserController | take\_screenshot() | 2 |
| **Metodi di Storage** |  |  |  |
| capture\_formatted\_response() | ResultsViewer | capture\_html\_response() | 1 |
| load\_analysis\_results() | ResultsViewer | display\_results() | 1 |
| format\_analysis\_results\_html() | ResultsViewer | format\_results\_html() | 1 |
| save\_response\_to\_project() | StorageManager | save\_response() | 1 |
| export\_to\_docx() | StorageManager | export\_as\_docx() | 1 |
| export\_to\_pdf() | StorageManager | export\_as\_pdf() | 1 |
| export\_to\_txt() | StorageManager | export\_as\_txt() | 1 |
| **Metodi di DB** |  |  |  |
| recupera\_ultimo\_progetto() | DatabaseManager | get\_last\_project() | 3 |
| ripristina\_ultima\_analisi() | DatabaseManager | restore\_last\_analysis() | 3 |
| ripristina\_analisi\_da\_database() | DatabaseManager | restore\_analysis\_from\_db() | 3 |
| load\_project\_details() | DatabaseManager | get\_project\_details() | 3 |
| diagnose\_and\_fix\_database() | DatabaseManager | repair\_database() | 3 |
| **Metodi di Generazione** |  |  |  |
| \_generate\_book\_crisp() | BookGenerator | generate\_book\_crisp() | 5 |
| \_generate\_book\_legacy() | BookGenerator | generate\_book\_legacy() | 5 |
| generate\_book() | BookGenerator | generate\_book() | 5 |
| \_parse\_book\_index() | BookGenerator | parse\_index() | 5 |
| \_load\_chapter\_prompt() | BookGenerator | load\_chapter\_prompt() | 5 |
| \_save\_chapter() | BookGenerator | save\_chapter() | 5 |

**4. Strategia per la Compatibilità Retroattiva**

Per garantire che il codice esistente continui a funzionare durante la rifattorizzazione, verrà utilizzato un approccio di "Facade Pattern":

[Usare una citazione significativa del documento per attirare l'attenzione del lettore o usare questo spazio per enfatizzare un punto chiave. Per posizionare questa casella di testo in un punto qualsiasi della pagina, è sufficiente trascinarla.]

Codice:

class AIBookBuilder:

"""

Classe Facade che mantiene l'API pubblica originale

ma delega il lavoro alle nuove classi specializzate

"""

def \_\_init\_\_(self):

# Inizializza il sistema rifattorizzato

self.app = PubliScriptApp()

self.event\_bus = self.app.event\_bus

# Mantiene riferimenti interni per retrocompatibilità

self.browser\_controller = self.app.browser\_controller

self.ui\_manager = self.app.ui\_manager

self.analysis\_manager = self.app.analysis\_manager

self.storage = self.app.storage

# Migra gli attributi essenziali

self.driver = None # Sarà gestito da browser\_controller

self.log\_history = [] # Gestito da Logger

self.chat\_manager = ChatManager(parent=self) # Mantieni per compatibilità

# ... altri attributi originali per compatibilità ...

def analyze\_market(self, book\_type, keyword, language, market, analysis\_prompt=None, use\_crisp=None):

"""

Esempio di metodo facade che delega alle nuove classi

mantenendo la stessa firma/interfaccia

"""

# Prepara i parametri

params = {

'book\_type': book\_type,

'keyword': keyword,

'language': language,

'market': market,

'analysis\_prompt': analysis\_prompt,

'use\_crisp': use\_crisp if use\_crisp is not None else self.use\_crisp

}

# Delega al nuovo componente

result = self.analysis\_manager.run\_analysis(\*\*params)

# Mantieni il comportamento originale di ritornare il log

return self.chat\_manager.get\_log\_history\_string()

# Altri metodi facade similarmente implementati...

**5. Tabella delle Dipendenze da Aggiornare**

| **File Dipendente** | **Tipo di Dipendenza** | **Strategia di Aggiornamento** | **Priorità** |
| --- | --- | --- | --- |
| main.py | Creazione istanza AIBookBuilder | Nessuna modifica necessaria con Facade | Bassa |
| app\_launcher.py | Utilizzo di AIBookBuilder.create\_interface() | Nessuna modifica necessaria con Facade | Bassa |
| chat\_manager.py | Riferimento a metodi AIBookBuilder | Aggiornare per utilizzare EventBus | Media |
| crisp\_framework.py | Utilizzo di AIBookBuilder per analisi | Gradualmente migrare a AnalysisManager | Alta |
| market\_analysis.py | Chiamate a AIBookBuilder per browser | Gradualmente migrare a BrowserController | Alta |
| ui/\*.py | Riferimenti all'interfaccia | Nessuna modifica necessaria con Facade | Bassa |

**6. Piano di Test**

| **Fase** | **Componenti da Testare** | **Strategia di Test** | **Criteri di Successo** |
| --- | --- | --- | --- |
| 1 | StorageManager, ResultsViewer | Test unitari + Test manuale UI | Visualizzazione corretta dei risultati |
| 2 | BrowserController | Test di integrazione con Selenium | Connessione e comunicazione stabile |
| 3 | AnalysisManager | Test funzionali per ogni tipo di analisi | Risultati corrispondenti a pre-rifattorizzazione |
| 4 | UIManager | Test manuali UI | UI responsiva e aggiornamenti visibili |
| 5 | Sistema completo | Test end-to-end | Workflow completo funzionante |

**7. Tempistiche e Milestones**

| **Milestone** | **Fase** | **Data Target** | **Risultato Atteso** |
| --- | --- | --- | --- |
| M1: Visualizzazione Risultati | 1 | Settimana 2 | Visualizzazione HTML funzionante |
| M2: Gestione Browser Robusta | 2 | Settimana 4 | Comunicazione browser stabile |
| M3: Analisi Modulare | 3 | Settimana 7 | Analisi di mercato rifattorizzata |
| M4: UI Reattiva | 4 | Settimana 9 | Interfaccia utente migliorata |
| M5: Sistema Completo | 5 | Settimana 12 | Migrazione completata |

**8. Diagramma di Sequenza per l'Analisi del Mercato (Post-Rifattorizzazione)**

┌──────┐ ┌────────────┐ ┌──────────────┐ ┌──────────┐ ┌───────────┐

│ User │ │AIBookBuilder│ │AnalysisManager│ │BrowserCtrl│ │ResultsView│

└──┬───┘ └──────┬─────┘ └───────┬──────┘ └─────┬────┘ └─────┬─────┘

│ │ │ │ │

│ analyze\_market() │ │ │ │

│────────────────────>│ │ │ │

│ │ │ │ │

│ │ run\_analysis() │ │ │

│ │───────────────────────>│ │ │

│ │ │ │ │

│ │ │ send\_message() │ │

│ │ │──────────────────────>│ │

│ │ │ │ execute in browser │

│ │ │ │ ───────────────────> │

│ │ │ │ │

│ │ │ │ capture\_response() │

│ │ │ │ <─────────────────── │

│ │ │ │ │

│ │ │ get\_response() │ │

│ │ │<─────────────────────-│ │

│ │ │ │ │

│ │ │ │ │

│ │ │────────────┐ │ │

│ │ │ │ process │ │

│ │ │<───────────┘ results │ │

│ │ │ │ │

│ │ │ display\_results() │ │

│ │ │──────────────────────────────────────────────>│

│ │ │ │ │

│ │ │ │ │

│ │ results ready │ │ │

│ │<─────────────────────--│ │ │

│ │ │ │ │

│ analysis complete │ │ │ │

│<────────────────────│ │ │ │

│ │ │ │ │

┌──┴───┐ ┌──────┴─────┐ ┌───────┴──────┐ ┌─────┴────┐ ┌─────┴─────┐

│ User │ │AIBookBuilder│ │AnalysisManager│ │BrowserCtrl│ │ResultsView│

└──────┘ └────────────┘ └──────────────┘ └──────────┘ └───────────┘

**9. Considerazioni sui Rischi**

| **Rischio** | **Probabilità** | **Impatto** | **Mitigazione** |
| --- | --- | --- | --- |
| Comportamento diverso dopo rifattorizzazione | Media | Alto | Test approfonditi per ogni fase |
| Tempi di implementazione lunghi | Alta | Medio | Approccio incrementale e prioritizzazione |
| Dipendenze circolari tra moduli | Alta | Alto | Design attento delle interfacce tra componenti |
| Resistenza al cambiamento | Media | Medio | Coinvolgimento team e documentazione chiara |
| Bug di regressione | Alta | Alto | Test di regressione automatizzati |

Questo piano dettagliato dovrebbe fornire una roadmap chiara per la rifattorizzazione di AIBookBuilder, mantenendo la compatibilità con il codice esistente mentre si migra gradualmente verso un'architettura più modulare e manutenibile.