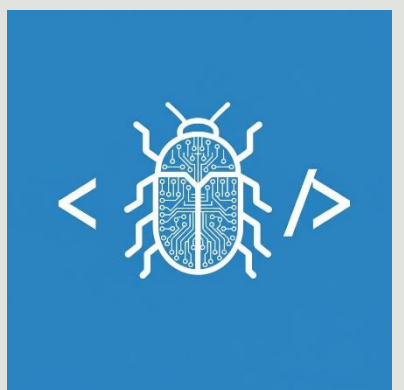


UNIVERSITÀ  
DEGLI STUDI DI  
PADOVA

# TECNOLOGIE

**GRUPPO 4**  
**BugBusters**

**Team:** Alberto Autiero, Marco Favero, Alberto Pignat, Marco Piro, Linor Sadè, Leonardo Salviato, Luca Slongo



# PANORAMICA

NEXUM è una piattaforma HR (Human Resources) sviluppata da Eggon in collaborazione con BugBusters (Gruppo 4 del corso di Ingegneria del Software, dell'Università degli Studi di Padova)

Obiettivo: Validare moduli sperimentali basati su AI e data analytics

## **Due ambiti funzionali principali:**

### 1. AI Assistant Generativo

→ Creazione automatizzata di contenuti per comunicazioni interne

### 2. AI Co-Pilot per Consulenti del Lavoro

→ Riconoscimento, split e dispaccio massivo di documenti (cedolini)



## AWS Bedrock

- Modello Amazon Nova Lite v1 per generazione testo e OCR
- **AI Assistant Generativo**
  - Converse API per generazione testo conversazionale
  - System prompt personalizzato con descrizione azienda e tono
  - Mantenimento contesto conversazione multi-turn
  - Generazione comunicazioni interne
- **AI Co-Pilot Document Analysis**
  - OCR e analisi documenti (PDF, PNG, JPEG)
  - Estrazione dati strutturati: tipo documento, dipendente, importo, date
  - Output JSON validato per split e dispaccio automatico



## Frontend - Angular

- **Framework principale**
  - Per dashboard amministrativa
  - Componenti modulari: AI Assistant, AI Co-Pilot, Analytics, Storico
- **RxJS 7.8.0**
  - Gestione asincrona chiamate API al backend
  - Polling per status documenti in analisi



## Backend - Ruby on Rails

- **API REST**
  - Endpoint per generazione testo AI (/genera)
  - Gestione documenti: upload, analisi, consultazione
  - Gestione conversazioni e storico messaggi
- **Service Pattern**
  - AiService: orchestrazione generazione testo con contesto aziendale
  - DocumentAnalysisService: OCR e estrazione dati da PDF/immagini
  - AiTextGenerator: adapter per AWS Bedrock SDK
- **Active Storage**
  - Gestione upload e storage documenti



## Database - PostgreSQL

- **Struttura dati relazionale**
  - Company: aziende che utilizzano la piattaforma
  - Conversation + Message: storico conversazioni AI con contesto
  - Tone: toni comunicativi personalizzati per azienda
  - Document: metadati documenti con status (pending/processing/completed)
- **Active Record ORM**
  - Relazioni: Company has\_many Documents, Conversations, Tones
  - Validazioni, enum per status, callbacks lifecycle
  - JSON column (ai\_data) per salvare dati estratti da Bedrock

# Architettura del Sistema

## FLUSSO DATI COMPLETO:

### 1. Frontend Angular

- Interfaccia utente con Angular Material
- Chiamate HTTP (RxJS) al backend Rails

### 2. Backend Rails API

- Controller gestiscono richieste HTTP
- Service orchestrano logica business
- Active Record per persistenza su PostgreSQL

### 3. AI Services

- AiService chiama AWS Bedrock Converse API
- DocumentAnalysisService processa documenti