**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**

Engenharia Informática e de Computadores

**Projecto e Seminário**

**Proposta de Projecto**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Membros do Grupo** | | | | | |
| **26657** | **Ricardo Neto** | **31923** | **Nuno Sousa** | **32223** | **Paulo Pires** |

1. **Introdução**

Com o aparecimento de software cada vez mais barato, torna-se necessário para as software houses reduzirem os custos de desenvolvimento de produtos, no sentido de conseguirem competir no mercado.

Os meios para chegar a esse fim dividem-se em diminuir o nível de exigência dos seus recursos humanos ou diminuir o ciclo de desenvolvimento do produto.

Com o presente projecto pretende-se criar uma solução que permita o desenvolvimento de aplicações de forma mais rápida, diminuindo assim todo o ciclo de desenvolvimento incluído.

1. **Análise**
   1. **Abordagem**

Considerando o ciclo de desenvolvimento de uma aplicações destacam-se as seguintes fases:

**No caminho para a fase de desenvolvimento, são produzidos outputs suficientes para se conseguir descrever um problema com algum detalhe.**

**Pretende-se que, ao estruturar essa informação num formato específico, seja possível codificar estruturas sobre as quais a equipa de desenvolvimento iniciará o seu trabalho.**

1. **Arquitectura**
   1. **Ficheiro descritor do problema**

**A presente solução visa criar regras para a construção de um ficheiro com meta-informação, que incluirá todos os outputs, descrevendo assim os vários elementos a serem codificados.**

**De toda informação passível de estar presente neste ficheiro, destacamos os seguintes elementos:**

|  |  |
| --- | --- |
| ****Tipo**** | ****Descrição**** |
| ****Environments**** | **Definição de servidores de base de dados, aplicacionais, e-mail, ftp, LDAP,...** |
| ****Tipos**** | **Gestão de tipos, tendo disponíveis por omissão os tipos primitivos básicos.** |
| ****Proxies**** | **Criação de entidades passando pela definição de atributos e seus domínios à relação entre entidades e sua cardinalidade e persistência.** |
| ****Processos**** | **Definição de processos com determinação de entidades envolvidas e regras a aplicar.** |
| ****Permissões**** | **Criação de matriz de permissões baseada em RBAC com vista à sua aplicação aos processos definidos.** |

* 1. **Módulo de Coordenação**

Com vista a permitir que o processo de codificação possa ter várias linguagens alvo, irá ser criado um módulo de coordenação que terá a função de orquestrar todo o processo.

Ao verificar o tipo de geração (e.g. Java, .NET, Ruby, SQL, MySQL...) indicada no ficheiro descritor, o coordenador irá solicitar a criação de uma instância desse gerador, recolher informação do ficheiro, passando-a ao gerador, através de uma interface bem definida.

* 1. **Gerador Automático de Código**

A possibilidade de geração de código em várias linguagens será suportada, naturalmente, por contrato, criando uma interface bem definida que definirá o comportamento que um gerador deverá ter.

O gerador terá que disponibilizar um factory que permita ao coordenador obter uma instância para o mesmo.

1. **Estrutura de Execução**

O plano de projecto está divido nas seguintes fases:

* Start-Up & Planning
* Execution
* Close-Down

As três fases descritas acima são compostas por 4 Work Packages (WP), que por sua vez estão dividas em várias tarefas e sub-tarefas.

Cada uma das WP contém um numero limitado de tarefas, permitindo assim um acompanhamento mais estruturado do trabalho efectuado. Nesta secção é apresentado o plano de trabalho – WP, para o desenvilvimento do Projecto final de curso.

* 1. **WP1 – Definição e elaboração da proposta de projecto**

O objectivo deste WP é definir e elaborar uma proposta de projecto final de curso.

As actividades decorrentes desta WP contemplam:

* Definição do âmbito projecto
* Planeamento de execução
  + 1. **Entregáveis do WP**
* Documento de proposta de projecto
  1. **WP2 – Implementação**

Esta fase do projecto é crucial para o sucesso do mesmo. Nesta, as características da solução a desenvolver serão definidos.

|  |  |  |  |
| --- | --- | --- | --- |
| Tarefas | Inputs | | Outputs |
| Definição de dicionário de dados | Documento de proposta de projecto | Dicionário de dados | |
| Extensões - Visual Studio |  | Costumização - Visual Studio | |
| Criação da infra-estrutura base (object-model) |  | Assemblies com a infra-estrutura base da solução | |
| Implementação de gerador baseado em dicionário de dados |  |  | |

* 1. **WP3 – Integração**

Nesta WP efectuar-se-à a integração das várias peças de software produzidas na WP2, bem como a definição e realização dos testes de integração.

|  |  |  |  |
| --- | --- | --- | --- |
| Tarefas | Inputs | | Outputs |
| Integração | Conjunto do software produzido na WP2 | Solução produzida na WP2 e revista na WP3 | |
| Definição e realização dos testes de aceitação |  | Resultado dos testes de integração | |

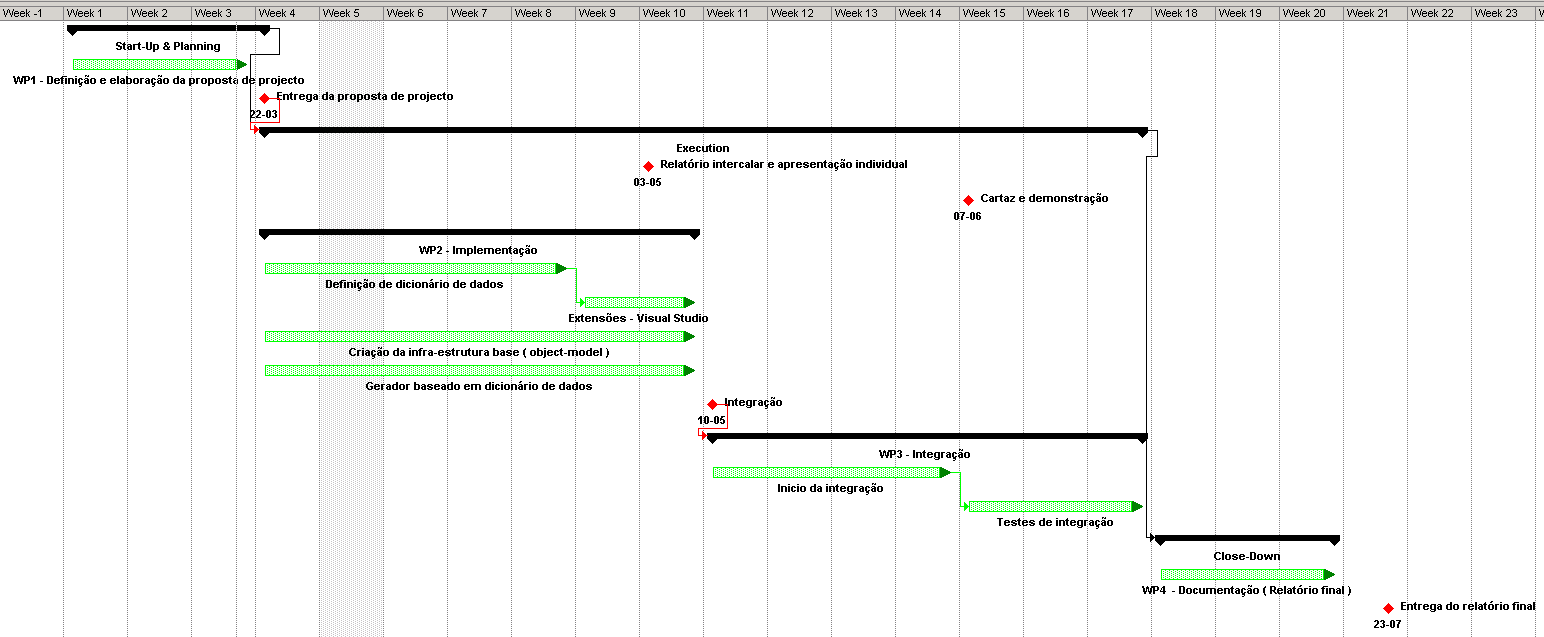
* 1. **WP4 – Documentação (Relatório Final)**

Nesta WP será materializado o relatório final.

|  |  |  |  |
| --- | --- | --- | --- |
| Tarefas | Inputs | | Outputs |
| Documentação da solução. ( relatório final ) | Documento de proposta de projecto | Relatório final | |
|  | Processos definidos na fase de implementação |  | |

* 1. **Plano de Projecto**

Considerando o faseamento do projecto (WP) definidos acima, resulta o plano de projecto apresentado na Figura abaixo, que detalha a distribuição do projecto no tempo.



* 1. **Marcos de Projecto**

|  |  |
| --- | --- |
| Milestones | Semanas de Trabalho |
| Entrega da da proposta de projecto | W 4 |
| Inicio da implementação | W 5 |
| Integração | W 11 |
| Cartaz e demonstração | W15 |
| Entrega do relatório final | 21 |