

SERPRO: Serviço Federal de Processamento de Dados

Daniel Miranda
Diego Martos Buoro
Mateus Barros Rodrigues
Renato Cordeiro Ferreira
Ruan de Menezes Costa
Thiago Gomes Toledo
Vinicius Vendramini
Wagner Ferreira Alves
Felipe Blassioli

MAC0332 - Engenharia de Software

Instituto de Matemática e Estatística (USP)

Novembro 2014

Sumário

- 1 Empresa
- 2 Sistemas
- 3 Processos
- 4 Desenvolvimento
- 5 Universidade
- 6 Conclusão



Fonte:

<http://g1.globo.com/concursos-e-emprego/noticia/2012/12/serpro-fara-concurso-para-360-vagas-de-nivel-medio-e-superior.html>

Dados

- Serviço Federal de Processamento de Dados
- 50 anos (fundado em 01/12/1964)
- 11.000 funcionários
- 4.000 da área de desenvolvimento
- 11 polos de integração (centros de comando) por todo o Brasil

Cientes

- Muitos clientes (federais, estaduais e municipais)
 - Receita Federal
 - PGFN (Procuradoria Geral da Fazenda Nacional)
 - PGF (Procuradoria Geral Federal)
 - Denatran
 - Ministérios

Clientes

- Muitos clientes (federais, estaduais e municipais)
 - Receita Federal
 - PGFN (Procuradoria Geral da Fazenda Nacional)
 - PGF (Procuradoria Geral Federal)
 - Denatran
 - Ministérios

Funcionários

- Entrada via concurso
- Organização em times e setores
- Grande tempo de permanência na empresa
- Colaboração entre sedes e interna
- Uso intensivo de ferramentas de comunicação

Funcionários

- Entrada via concurso
- Organização em times e setores
- Grande tempo de permanência na empresa
- Colaboração entre sedes e interna
- Uso intensivo de ferramentas de comunicação

Funcionários

- Entrada via concurso
- Organização em times e setores
- Grande tempo de permanência na empresa
- Colaboração entre sedes e interna
- Uso intensivo de ferramentas de comunicação

Funcionários

- Entrada via concurso
- Organização em times e setores
- Grande tempo de permanência na empresa
- Colaboração entre sedes e interna
- Uso intensivo de ferramentas de comunicação

Funcionários

- Entrada via concurso
- Organização em times e setores
- Grande tempo de permanência na empresa
- Colaboração entre sedes e interna
- Uso intensivo de ferramentas de comunicação

Aspectos gerais

- **Sistemas de informação**
- Volume grande de informações
- Muitos sistemas críticos
- Cenários diversos de integração (online e batch)
- Servidores de alta e baixa plataforma
- Aplicações para desktop, web e mobile
- Aplicações web em Internet e Intranet

Aspectos gerais

- Sistemas de informação
- Volume grande de informações
- Muitos sistemas críticos
- Cenários diversos de integração (online e batch)
- Servidores de alta e baixa plataforma
- Aplicações para desktop, web e mobile
- Aplicações web em Internet e Intranet

Aspectos gerais

- Sistemas de informação
- Volume grande de informações
- Muitos sistemas críticos
- Cenários diversos de integração (online e batch)
- Servidores de alta e baixa plataforma
- Aplicações para desktop, web e mobile
- Aplicações web em Internet e Intranet

Aspectos gerais

- Sistemas de informação
- Volume grande de informações
- Muitos sistemas críticos
- Cenários diversos de integração (online e batch)
- Servidores de alta e baixa plataforma
- Aplicações para desktop, web e mobile
- Aplicações web em Internet e Intranet

Aspectos gerais

- Sistemas de informação
- Volume grande de informações
- Muitos sistemas críticos
- Cenários diversos de integração (online e batch)
- Servidores de alta e baixa plataforma
- Aplicações para desktop, web e mobile
- Aplicações web em Internet e Intranet

Aspectos gerais

- Sistemas de informação
- Volume grande de informações
- Muitos sistemas críticos
- Cenários diversos de integração (online e batch)
- Servidores de alta e baixa plataforma
- Aplicações para desktop, web e mobile
- Aplicações web em Internet e Intranet

Aspectos gerais

- Sistemas de informação
- Volume grande de informações
- Muitos sistemas críticos
- Cenários diversos de integração (online e batch)
- Servidores de alta e baixa plataforma
- Aplicações para desktop, web e mobile
- Aplicações web em Internet e Intranet

Sistemas - Linguagens

- Principal: Java
- Web: PHP, ASP, ASP.NET, VB.NET, Javascript
- Desktop: VB, C#, Delphi, Python
- Alta plataforma: Assembly, COBOL, NATURAL

Sistemas - Linguagens

- Principal: Java
- Web: PHP, ASP, ASP.NET, VB.NET, Javascript
- Desktop: VB, C#, Delphi, Python
- Alta plataforma: Assembly, COBOL, NATURAL

Sistemas - Linguagens

- Principal: Java
- Web: PHP, ASP, ASP.NET, VB.NET, Javascript
- Desktop: VB, C#, Delphi, Python
- Alta plataforma: Assembly, COBOL, NATURAL

Sistemas - Linguagens

- Principal: Java
- Web: PHP, ASP, ASP.NET, VB.NET, Javascript
- Desktop: VB, C#, Delphi, Python
- Alta plataforma: Assembly, COBOL, NATURAL

Desenvolvimento em Cascata

- Metodologia tradicional
- Certificação CMMI nível 3 (antiga)
- PSDS: Processo SERPRO de Desenvolvimento de Soluções
 - Baseado no CMM e RUP.
 - Flexibilizado em novas versões
 - Diretrizes: O que? Como? Quando? Por quem? Com o que?

Desenvolvimento em Cascata

- Metodologia tradicional
- Certificação CMMI nível 3 (antiga)
- PSDS: Processo SERPRO de Desenvolvimento de Soluções
 - Baseado no CMM e RUP.
 - Flexibilizado em novas versões
 - Diretrizes: O que? Como? Quando? Por quem? Com o que?

Desenvolvimento em Cascata

- Metodologia tradicional
- Certificação CMMI nível 3 (antiga)
- PSDS: Processo SERPRO de Desenvolvimento de Soluções
 - Baseado no CMM e RUP.
 - Flexibilizado em novas versões
 - Diretrizes: O que? Como? Quando? Por quem? Com o que?

Desenvolvimento em Cascata

- Metodologia tradicional
- Certificação CMMI nível 3 (antiga)
- PSDS: Processo SERPRO de Desenvolvimento de Soluções
 - Baseado no CMM e RUP.
 - Flexibilizado em novas versões
 - Diretrizes: O que? Como? Quando? Por quem? Com o que?

Desenvolvimento em Cascata

- Metodologia tradicional
- Certificação CMMI nível 3 (antiga)
- PSDS: Processo SERPRO de Desenvolvimento de Soluções
 - Baseado no CMM e RUP.
 - Flexibilizado em novas versões
 - Diretrizes: O que? Como? Quando? Por quem? Com o que?

Desenvolvimento em Cascata

- Metodologia tradicional
- Certificação CMMI nível 3 (antiga)
- PSDS: Processo SERPRO de Desenvolvimento de Soluções
 - Baseado no CMM e RUP.
 - Flexibilizado em novas versões
 - Diretrizes: O que? Como? Quando? Por quem? Com o que?

Desenvolvimento em Cascata

- Área de negócios
- Analista de Requisitos
- Analista de Desenvolvimento
- Analista de Testes
- Analista de Infraestrutura
- Administrador de Banco de Dados (vários níveis)
- Desenvolvedores

Desenvolvimento Ágil

- Início em vertente Ágil do PSDS
- Aproximação dos papéis tradicionais
- SCRUM
 - DevOps (Desenvolvedor e Operador)
 - PO (*Product Owner*)
 - Scrum master
- Principal a partir de 2015

Desenvolvimento Ágil

- Início em vertente Ágil do PSDS
- Aproximação dos papéis tradicionais
- SCRUM
 - DevOps (Desenvolvedor e Operador)
 - PO (*Product Owner*)
 - Scrum master
- Principal a partir de 2015

Desenvolvimento Ágil

- Início em vertente Ágil do PSDS
- Aproximação dos papéis tradicionais
- SCRUM
 - DevOps (Desenvolvedor e Operador)
 - PO (*Product Owner*)
 - Scrum master
- Principal a partir de 2015

Desenvolvimento Ágil

- Início em vertente Ágil do PSDS
- Aproximação dos papéis tradicionais
- SCRUM
 - DevOps (Desenvolvedor e Operador)
 - PO (*Product Owner*)
 - Scrum master
- Principal a partir de 2015

Desenvolvimento Ágil

- Início em vertente Ágil do PSDS
- Aproximação dos papéis tradicionais
- SCRUM
 - DevOps (Desenvolvedor e Operador)
 - PO (*Product Owner*)
 - Scrum master
- Principal a partir de 2015

Desenvolvimento Ágil

- Início em vertente Ágil do PSDS
- Aproximação dos papéis tradicionais
- SCRUM
 - DevOps (Desenvolvedor e Operador)
 - PO (*Product Owner*)
 - Scrum master
- Principal a partir de 2015

Requisitos

	Cascata (PSDS)	Ágil (SCRUM)
Mudanças	Reiniciam o processo e demandam renegociação	Trocam prioridades e funcionalidades
Documentação	Casos de uso + ALM (<i>Application Lifecycle Management</i>)	Histórias de usuário + Kanban
Bugs	ALM + Análise de Impacto	

Documentação técnica

- UML
 - Adaptações para sistemas não orientados a objetos
 - Ferramentas: Astah e Rational Rose
 - Frequência de uso:
 - Cascata: criação para todo o projeto
 - Ágil: criação focada na parte a ser desenvolvida

Documentação técnica

- UML
- Adaptações para sistemas não orientados a objetos
- Ferramentas: Astah e Rational Rose
- Frequência de uso:
 - Cascata: criação para todo o projeto
 - Ágil: criação focada na parte a ser desenvolvida

Documentação técnica

- UML
- Adaptações para sistemas não orientados a objetos
- Ferramentas: Astah e Rational Rose
- Frequência de uso:
 - Cascata: criação para todo o projeto
 - Ágil: criação focada na parte a ser desenvolvida

Documentação técnica

- UML
- Adaptações para sistemas não orientados a objetos
- Ferramentas: Astah e Rational Rose
- Frequência de uso:
 - Cascata: criação para todo o projeto
 - Ágil: criação focada na parte a ser desenvolvida

Arquitetura e Design

- Analistas de sistema e arquitetos de software
- UML
- Padrões de projeto: MVC e GoF
- Múltiplos ambientes
 - Desenvolvimento
 - Testes
 - Homologação
 - Treinamento
 - Produtivo (em uso)

Arquitetura e Design

- Analistas de sistema e arquitetos de software
- UML
- Padrões de projeto: MVC e GoF
- Múltiplos ambientes
 - Desenvolvimento
 - Testes
 - Homologação
 - Treinamento
 - Produtivo (em uso)

Arquitetura e Design

- Analistas de sistema e arquitetos de software
- UML
- Padrões de projeto: MVC e GoF
- Múltiplos ambientes
 - Desenvolvimento
 - Testes
 - Homologação
 - Treinamento
 - Produtivo (em uso)

Arquitetura e Design

- Analistas de sistema e arquitetos de software
- UML
- Padrões de projeto: MVC e GoF
- Múltiplos ambientes
 - Desenvolvimento
 - Testes
 - Homologação
 - Treinamento
 - Produtivo (em uso)

Testes Automatizados

- Não obrigatórios (decisão da equipe)
- Caixa branca
- Metodologia livre (TDD, BDD)
- 3 tipos:
 - de Unidade (com *mocks* ou *stubs*)
 - Integração (com acesso a ferramentas externas)
 - Aceitação (com DBehave ou Selenium)

Testes Automatizados

- Não obrigatórios (decisão da equipe)
- Caixa branca
- Metodologia livre (TDD, BDD)
- 3 tipos:
 - de Unidade (com *mocks* ou *stubs*)
 - Integração (com acesso a ferramentas externas)
 - Aceitação (com DBehave ou Selenium)

Testes Automatizados

- Não obrigatórios (decisão da equipe)
- Caixa branca
- Metodologia livre (TDD, BDD)
- 3 tipos:
 - de Unidade (com *mocks* ou *stubs*)
 - Integração (com acesso a ferramentas externas)
 - Aceitação (com DBehave ou Selenium)

Testes Automatizados

- Não obrigatórios (decisão da equipe)
- Caixa branca
- Metodologia livre (TDD, BDD)
- 3 tipos:
 - de Unidade (com *mocks* ou *stubs*)
 - Integração (com acesso a ferramentas externas)
 - Aceitação (com DBehave ou Selenium)

Testes Funcionais

- Obrigatórios (analistas de testes)
- Caixa preta
- 2 tipos:
 - Estáticos (poucos, fora de execução)
 - Dinâmicos (maioria, em execução)

Testes Funcionais

- Obrigatórios (analistas de testes)
- Caixa preta
- 2 tipos:
 - Estáticos (poucos, fora de execução)
 - Dinâmicos (maioria, em execução)

Testes Funcionais

- Obrigatórios (analistas de testes)
- Caixa preta
- 2 tipos:
 - Estáticos (poucos, fora de execução)
 - Dinâmicos (maioria, em execução)

Métricas

- Quantidade de erros identificados durante os testes
- Quantidade de correções efetuadas
- Densidade de defeitos no ambiente de Homologação (através do ALM)
- Eficiência na detecção de defeitos
- Desafio: como medir a cobertura com relação aos requisitos?

Métricas

- Quantidade de erros identificados durante os testes
- Quantidade de correções efetuadas
- Densidade de defeitos no ambiente de Homologação (através do ALM)
- Eficiência na detecção de defeitos
- Desafio: como medir a cobertura com relação aos requisitos?

Métricas

- Quantidade de erros identificados durante os testes
- Quantidade de correções efetuadas
- Densidade de defeitos no ambiente de Homologação (através do ALM)
- Eficiência na detecção de defeitos
- Desafio: como medir a cobertura com relação aos requisitos?

Métricas

- Quantidade de erros identificados durante os testes
- Quantidade de correções efetuadas
- Densidade de defeitos no ambiente de Homologação (através do ALM)
- Eficiência na detecção de defeitos
- Desafio: como medir a cobertura com relação aos requisitos?

Métricas

- Quantidade de erros identificados durante os testes
- Quantidade de correções efetuadas
- Densidade de defeitos no ambiente de Homologação (através do ALM)
- Eficiência na detecção de defeitos
- Desafio: como medir a cobertura com relação aos requisitos?

Ferramentas e versionamento

- Principal: ALM (*Application Lifecycle Management*) da IBM
 - Requisitos, código fonte, relatório de testes e falhas
 - União de várias etapas em um sistema
- Sistemas Operacionais: Ubuntu (Linux), Windows
- Frameworks: Demoiselle (web, open source)
- IDEs: Eclipse, Visual Studio, Emulador 3270 (mainframes)
- Versionamento: CVS, ALM (Jazz)
- Padronização: Checkstyle, PMD, FindBugs (SONAR)
- Bancos de Dados: Oracle (principal), PostgreSQL
- Integração contínua: JBE (Jazz Build Engine), Continuum e Jenkins
- Testes Funcionais: DBehave e Selenium

Ferramentas e versionamento

- Principal: ALM (*Application Lifecycle Management*) da IBM
 - Requisitos, código fonte, relatório de testes e falhas
 - União de várias etapas em um sistema
- Sistemas Operacionais: Ubuntu (Linux), Windows
- Frameworks: Demoiselle (web, open source)
- IDEs: Eclipse, Visual Studio, Emulador 3270 (mainframes)
- Versionamento: CVS, ALM (Jazz)
- Padronização: Checkstyle, PMD, FindBugs (SONAR)
- Bancos de Dados: Oracle (principal), PostgreSQL
- Integração contínua: JBE (Jazz Build Engine), Continuum e Jenkins
- Testes Funcionais: DBehave e Selenium

Ferramentas e versionamento

- Principal: ALM (*Application Lifecycle Management*) da IBM
 - Requisitos, código fonte, relatório de testes e falhas
 - União de várias etapas em um sistema
- Sistemas Operacionais: Ubuntu (Linux), Windows
- Frameworks: Demoiselle (web, open source)
- IDEs: Eclipse, Visual Studio, Emulador 3270 (mainframes)
- Versionamento: CVS, ALM (Jazz)
- Padronização: Checkstyle, PMD, FindBugs (SONAR)
- Bancos de Dados: Oracle (principal), PostgreSQL
- Integração contínua: JBE (Jazz Build Engine), Continuum e Jenkins
- Testes Funcionais: DBehave e Selenium

Ferramentas e versionamento

- Principal: ALM (*Application Lifecycle Management*) da IBM
 - Requisitos, código fonte, relatório de testes e falhas
 - União de várias etapas em um sistema
- Sistemas Operacionais: Ubuntu (Linux), Windows
- Frameworks: Demoiselle (web, open source)
- IDEs: Eclipse, Visual Studio, Emulador 3270 (mainframes)
- Versionamento: CVS, ALM (Jazz)
- Padronização: Checkstyle, PMD, FindBugs (SONAR)
- Bancos de Dados: Oracle (principal), PostgreSQL
- Integração contínua: JBE (Jazz Build Engine), Continuum e Jenkins
- Testes Funcionais: DBehave e Selenium

Ferramentas e versionamento

- Principal: ALM (*Application Lifecycle Management*) da IBM
 - Requisitos, código fonte, relatório de testes e falhas
 - União de várias etapas em um sistema
- Sistemas Operacionais: Ubuntu (Linux), Windows
- Frameworks: Demoiselle (web, open source)
- IDEs: Eclipse, Visual Studio, Emulador 3270 (mainframes)
- Versionamento: CVS, ALM (Jazz)
- Padronização: Checkstyle, PMD, FindBugs (SONAR)
- Bancos de Dados: Oracle (principal), PostgreSQL
- Integração contínua: JBE (Jazz Build Engine), Continuum e Jenkins
- Testes Funcionais: DBehave e Selenium

Ferramentas e versionamento

- Principal: ALM (*Application Lifecycle Management*) da IBM
 - Requisitos, código fonte, relatório de testes e falhas
 - União de várias etapas em um sistema
- Sistemas Operacionais: Ubuntu (Linux), Windows
- Frameworks: Demoiselle (web, open source)
- IDEs: Eclipse, Visual Studio, Emulador 3270 (mainframes)
- Versionamento: CVS, ALM (Jazz)
- Padronização: Checkstyle, PMD, FindBugs (SONAR)
- Bancos de Dados: Oracle (principal), PostgreSQL
- Integração contínua: JBE (Jazz Build Engine), Continuum e Jenkins
- Testes Funcionais: DBehave e Selenium

Ferramentas e versionamento

- Principal: ALM (*Application Lifecycle Management*) da IBM
 - Requisitos, código fonte, relatório de testes e falhas
 - União de várias etapas em um sistema
- Sistemas Operacionais: Ubuntu (Linux), Windows
- Frameworks: Demoiselle (web, open source)
- IDEs: Eclipse, Visual Studio, Emulador 3270 (mainframes)
- Versionamento: CVS, ALM (Jazz)
- Padronização: Checkstyle, PMD, FindBugs (SONAR)
- Bancos de Dados: Oracle (principal), PostgreSQL
- Integração contínua: JBE (Jazz Build Engine), Continuum e Jenkins
- Testes Funcionais: DBehave e Selenium

Ferramentas e versionamento

- Principal: ALM (*Application Lifecycle Management*) da IBM
 - Requisitos, código fonte, relatório de testes e falhas
 - União de várias etapas em um sistema
- Sistemas Operacionais: Ubuntu (Linux), Windows
- Frameworks: Demoiselle (web, open source)
- IDEs: Eclipse, Visual Studio, Emulador 3270 (mainframes)
- Versionamento: CVS, ALM (Jazz)
- Padronização: Checkstyle, PMD, FindBugs (SONAR)
- Bancos de Dados: Oracle (principal), PostgreSQL
- Integração contínua: JBE (Jazz Build Engine), Continuum e Jenkins
- Testes Funcionais: DBehave e Selenium

Ferramentas e versionamento

- Principal: ALM (*Application Lifecycle Management*) da IBM
 - Requisitos, código fonte, relatório de testes e falhas
 - União de várias etapas em um sistema
- Sistemas Operacionais: Ubuntu (Linux), Windows
- Frameworks: Demoiselle (web, open source)
- IDEs: Eclipse, Visual Studio, Emulador 3270 (mainframes)
- Versionamento: CVS, ALM (Jazz)
- Padronização: Checkstyle, PMD, FindBugs (SONAR)
- Bancos de Dados: Oracle (principal), PostgreSQL
- Integração contínua: JBE (Jazz Build Engine), Continuum e Jenkins
- Testes Funcionais: DBehave e Selenium

Ambiente de desenvolvimento



Ambiente de desenvolvimento



Ambiente de desenvolvimento



Ambiente de desenvolvimento



Evolução

- Abundância de software legado
- Grande quantidade de plataformas e infraestrutura
- Substituição completa ou em paralelo
- Integração de sistemas legados com mais novos
- Replicação, proteção e evolução de dados

Evolução

- Abundância de software legado
- Grande quantidade de plataformas e infraestrutura
- Substituição completa ou em paralelo
- Integração de sistemas legados com mais novos
- Replicação, proteção e evolução de dados

Evolução

- Abundância de software legado
- Grande quantidade de plataformas e infraestrutura
- Substituição completa ou em paralelo
- Integração de sistemas legados com mais novos
- Replicação, proteção e evolução de dados

Evolução

- Abundância de software legado
- Grande quantidade de plataformas e infraestrutura
- Substituição completa ou em paralelo
- Integração de sistemas legados com mais novos
- Replicação, proteção e evolução de dados

Evolução

- Abundância de software legado
- Grande quantidade de plataformas e infraestrutura
- Substituição completa ou em paralelo
- Integração de sistemas legados com mais novos
- Replicação, proteção e evolução de dados

Suporte e Manutenção





Fonte: https://www.serpro.gov.br/noticias/serpro-moderniza-centro-de-comando-em-sao-paulo/image_large

Evolução

- Programas de bolsa de estudo
- Liberação para palestras e horários
- Boa relação com os recém-formados
 - Novas tecnologias e ideias
 - Menos vícios
 - Possibilidade de destaque

Evolução

- Programas de bolsa de estudo
- Liberação para palestras e horários
- Boa relação com os recém-formados
 - Novas tecnologias e ideias
 - Menos vícios
 - Possibilidade de destaque

Evolução

- Programas de bolsa de estudo
- Liberação para palestras e horários
- Boa relação com os recém-formados
 - Novas tecnologias e ideias
 - Menos vícios
 - Possibilidade de destaque

Aspectos Gerais

- Pontos positivos
- Possíveis melhorias
- Sucessos e fracassos

Aspectos Gerais

- Pontos positivos
- Possíveis melhorias
- Sucessos e fracassos

Aspectos Gerais

- Pontos positivos
- Possíveis melhorias
- Sucessos e fracassos

SERPRO: Serviço Federal de Processamento de Dados

Daniel Miranda
Diego Martos Buoro
Mateus Barros Rodrigues
Renato Cordeiro Ferreira
Ruan de Menezes Costa
Thiago Gomes Toledo
Vinicius Vendramini
Wagner Ferreira Alves
Felipe Blassioli

MAC0332 - Engenharia de Software

Instituto de Matemática e Estatística (USP)

Novembro 2014