

Descrição do Trabalho – Airbnb

Objetivos

1. Conhecer e praticar as técnicas de projeto e análise de sistemas usando UML
2. Trabalhar em grupo para transformar uma descrição vaga do problema em um relatório e modelos detalhados e de alta qualidade

Grupos

- Serão seis grupos de **quatro a seis componentes**, dependendo do número de inscritos na turma. Antes do final do SAID, nenhum grupo de seis poderá ser formado.
- Forme o grupo de forma que pelo menos um aluno possa levar um **laptop para todas as aulas de acompanhamento**.
- Alunos que chegarem atrasados no curso e não encontrarem um grupo, serão alocados em um novo grupo, se menos de seis grupos tiverem sido formados. Caso contrário, terão que unir-se a um grupo existente.
- Exceções não serão abertas. É preciso saber trabalhar em grupo.

Entrega

- Entregas incrementais através do Moodle da disciplina. Prazos de cada incremento já estão definidos no Moodle, mas poderão mudar dependendo do decorrer da disciplina. Mudanças de prazo serão anunciadas em aula e através do fórum.
- A cada entrega, os alunos devem submeter **um único documento contendo todas as partes entregues até o momento**. Isso é importante para verificar a consistência entre os diagramas.
- Alunos terão de entre 10 e 32 dias a partir da aula da matéria correspondente para entregar o incremento do trabalho. Muitas vezes, as entregas serão semanais.
- Data da entrega do trabalho final: 29/11/2015. Uma cópia impressa do trabalho, sem modificações, deve ser entregue no dia 01/12/2015. Datas podem mudar em função do calendário da UERJ e do andamento das aulas. Mudanças de datas serão anunciadas no fórum da disciplina, mas **NÃO serão atualizadas neste documento**.

Organização do trabalho & Orientação

- Várias aulas serão dedicadas ao trabalho, mas horas adicionais serão necessárias.
- Serão feitas reuniões de acompanhamento com os grupos para apresentar e discutir o andamento do trabalho. **A presença de todos os membros do grupo é obrigatória**. Aluno pode ser retirado do grupo se perder mais do que 25% das reuniões de acompanhamento. Feedback não será dado por e-mail.
- Presença nas aulas dedicadas ao trabalho será cobrada normalmente.
- O tempo de acompanhamento é limitado. Portanto, atrasos do grupo não serão compensados para não prejudicar os outros grupos. A primeira reunião de acompanhamento começará às 19:30. Portanto, planejem-se para chegar no

horário.

Visão Geral do Trabalho

- Grupos devem partir de um sistema real (<https://www.airbnb.com.br/>) e propor uma extensão para o mesmo. O sistema com a extensão proposta será modelado durante o semestre.

Descrição Detalhada das Entregas:

Atenção: Está é uma descrição preliminar das entregas. Mudanças no enunciado podem ocorrer e serão anunciadas no fórum da disciplina. Mudanças nos enunciados NÃO serão atualizadas neste documento.

Os grupos podem aprimorar o trabalho até o prazo final. Ainda assim, os prazos das entregas incrementais devem ser respeitados. Atraso na entrega dos incrementos implica em perda de pontos (0.1 por atraso).

Entrega 1: Descrição da extensão proposta

- Descrição da extensão proposta para o Airbnb (<https://www.airbnb.com.br/>).
- Airbnb está se reinventando (<http://www.cnn.com/2014/05/12/airbnb-looks-beyond-flat-sharing-to-tickets-and-travel.html>). Pensem: uma plataforma para alugar quartos, casas e conectar pessoas poderia ser usada para oferecer que novos serviços? Estúdio de fotógrafos? Limpeza? Sejam criativos!
- A extensão deve utilizar **no mínimo 2 sistemas externos**, além do sistema de cartão de crédito.
- **Não são permitidas extensões de guias turísticos** para cidades para não ficar parecido com o trabalho do semestre passado.
- As extensões devem ser **suficientemente complexas** para justificar a necessidade de uma modelagem em UML.
- Importante: **A extensão proposta não pode ser parecida com nenhuma** de outro grupo. Portanto, leia as propostas existentes antes de submeter a sua. No caso de propostas parecidas, somente a primeira será aceita.
- Propostas de extensão devem ser submetidas através da tarefa no tópico “Trabalho do Curso”, na tarefa “Extensões do Airbnb.”
- O número do grupo deve ser colocado em função das entradas anteriores. Por exemplo, se já houver duas propostas na base de dados, o número do seu grupo será 3.

Entrega 2: Resumo, Introdução e Modelo de Contexto

- Resumo do Trabalho: Um ou dois parágrafos rapidamente descrevendo o Airbnb e a extensão e dizendo que o trabalho vai apresentar a modelagem parcial do sistema utilizando a UML
- Introdução: **Breve descrição do sistema original. Descrição detalhada da extensão proposta.** Que serviços a extensão oferece? Como o usuário interage com ela? Que opções ele tem? Quem mais interage com o sistema? Como e porque?
- Diagrama de contexto: Introdução ao diagrama de contexto com **a explicação do modelo**. Para que serve um diagrama de contexto? Com que outros sistemas

externos o sistema estendido do Airbnb tem que se comunicar? Mostrar o diagrama.

- Por favor, já criem documento com **título, índice e lista de diagramas**. Vejam a estrutura do trabalho de exemplo na página principal.

Entrega 3: Modelo de Casos de Uso

- **Diagrama de casos de uso para todo o sistema** do Airbnb, incluindo a extensão proposta.
- Descrição resumida de **todos os casos de uso e atores** do sistema: um parágrafo descrevendo o cenário de sucesso principal (conforme feito na análise de requisitos inicial)
- **Um caso de uso detalhado da extensão proposta**, utilizando o **template expandido** dado em sala (conforme feito na iteração que vai desenvolver o caso de uso). Deve conter todos os campos.
- Lembre-se de que cada diagrama deve ser precedido por um texto introduzindo e descrevendo o diagrama em detalhes, incluindo a justificativa da modelagem adotada. Não esquecer de explicar também o uso de cada **includes, extends e herança**.

Entrega 4: Diagrama de Classes de Análise

- **Diagrama de classes de análise** para o sistema com a extensão proposta. Representar classes, atributos, relacionamentos, multiplicidades. Nomear os relacionamentos. Classes não devem conter métodos, pois elas representam conceitos do mundo real, não objetos do sistema.
- Utilizar **lista de categorias ou análise gramatical** dos casos de uso para identificar as classes candidatas. Adicionar lista de categorias ou análise gramatical como Apêndice 1.
- **Classes em Java** para cada conceito do domínio identificado, com todos os atributos e sem nenhum método. **Atenção:** o diagrama de classes de análise representa os conceitos do domínio, não objetos do sistema. Só estou pedido as classes em Java para facilitar a implementação depois. Adicionar as classes em Java como Apêndice 2.
- Lembre-se de que cada diagrama deve ser precedido por um texto introduzindo e descrevendo o diagrama em detalhes, incluindo a justificativa da modelagem adotada. Não esquecer de explicar as classes e os **relacionamentos de composição, agregação e herança**.

Entrega 5: Diagrama de Sequência do Sistema e Diagrama de Pacotes

- Diagrama de **sequência do sistema** para cada fluxo do caso de uso estendido (principal, alternativo e de exceção), apresentado anteriormente.
- **Diagrama de pacotes** para o sistema com a extensão.
- Lembre-se de que cada diagrama deve ser precedido por um texto introduzindo e descrevendo-o em detalhes. No diagrama de pacotes, explicar camadas, partições e relacionamentos.

Entrega 6: Diagrama de Classes de Projeto e Diagrama de Classes de Sequência

- O **diagrama de classes de projeto** deve incluir:

- todas as classes necessárias para implementar o caso de uso detalhado da extensão de vocês
- todas as classes de IU
- todos os sistemas externos (vou comparar com o diagrama de contexto e de casos de uso)
- todas as interfaces do usuário (ex, android e Web)
- classes que permitam salvar objetos (em um banco de dados relacional ou em arquivos textos)
- classes que permitam falar com sistemas externos
- outras classes de suporte.
- Mostrar **atributos, métodos, multiplicidades, e nomear todos os relacionamentos**. Usar estereótipos para **identificar** classes de **fronteira, controle e entidade**.
- Os **diagramas de sequência** devem mostrar como as classes devem cooperar para atender o **fluxo principal e pelo menos um fluxo alternativo** do caso de uso detalhado. Aconselho separar os diagramas para facilitar a leitura. Usar **estereótipos** para identificar classes de **fronteira, controle e entidade**.
- **Implementação em Java**, funcionando, do fluxo principal. Incluir **instruções para rodar** e o código da implementação Java no Apêndice 3.
 - Simulem o comportamento de objetos complexos, quando necessário. Por exemplo, você podem criar um objeto simples que simula o seu sistema externo, ou seja, que fala com a sua fronteira. Ele pode retornar sempre os mesmos dados (que você espera para aquela interação). Você pode simular o objeto que salva em um banco de dados ou salvar objetos em um arquivo texto.
- Nesta entrega, mais do que nas outras, ambos os diagramas devem ser explicados em detalhes.
 - No diagrama de classes, descrever cada uma das **classes**, os **relacionamentos de herança, agregação e composição**, explicar porque as classes foram definidas como **fronteira, controle e entidade**, explicar como a definição das classes foram influenciadas pelos **padrões GRASP**.
 - No diagrama de sequência, explicar **quadros de interação, loops e condicionais**.
- Lembretes importantes:
 - Os diagramas de classe e de sequência devem ser consistentes. Vou comparar métodos, parâmetros, etc. Desenvolva-os em paralelo!
 - O sistema deve criar objetos. Quem cria cada objeto é uma decisão importante!

Entrega 7: Diagrama de Atividades, Diagrama de Estados e Conclusão

- Um **diagrama de atividades** representando um fluxo de trabalho (suficientemente complexo) do sistema. Fica a critério de vocês escolher que fluxo de trabalho. Pode ser, por exemplo, o fluxo de trabalho do caso de uso mais complexo, ou envolvendo mais de um caso de uso.
 - O diagrama de atividades deve incluir:
 - **raias de natação**
 - um ou mais **objetos**
 - **ramificações e ou bifurcações**
 - Explicar **atividades, ramificações ou bifurcações, e objetos**.

- **Diagramas de estados** de um **objeto complexo** do diagrama de classes de vocês.
 - O diagrama de estados deve incluir:
 - **ações** (dentro dos estados e/ou nas transições)
 - **mais de um tipo de evento**, se possível
 - Explicar **estados, eventos**, porque se definiu **uma ação dentro do estado ou na transição**, etc.
- **Conclusão**: resumo do que o trabalho apresentado. O que vocês aprenderam? Quais foram as maiores dificuldades?

Formato do Relatório

- Título do trabalho, nome dos componentes, data, disciplina, professora.
- Resumo do trabalho
- Índice
- Lista de Diagramas (com páginas)
- Resumo do trabalho
- Capítulo 1: Introdução
- Capítulo 2: Modelo de Contexto
- Capítulo 3: Modelo de Casos de Uso
- Capítulo 4: Modelo de Sequência do Sistema
- Capítulo 5: Modelo de Classes de Domínio
- Capítulo 6: Modelo de Pacotes
- Capítulo 7: Modelo de Classes de Projeto
- Capítulo 8: Modelo de Sequência de Projeto
- Capítulo 9: Modelo de Estados
- Capítulo 10: Modelo de Atividades
- Capítulo 11: Conclusão
- Referências (padrão ABNT)
- Apêndice 1: Lista de categorias ou análise gramatical do diagrama de classes.
- Apêndice 2: Código fonte das classes de domínio.
- Apêndice 3: Código fonte da implementação do diagrama de classes e de sequência. Instruções para rodar o código.

Texto claro e bem estruturado é fundamental.

Avaliação:

- Na terça-feira seguinte a entrega de cada incremento, os grupos deverão entregar uma avaliação da participação (de 0 a 100) de cada membro do grupo através do Moodle, assinada por todos (busquem a forma de que todos assinem – sem desculpas). Isso é para impedir que alunos que não fizeram nada ganhem a mesma nota do resto do grupo. Sejam justos e tenham autocrítica.
- Para evitar conflitos, sugiro que todos os membros do grupo se dediquem ao máximo durante o semestre.

Sugestão:

- **Não divida os modelos do trabalho entre os componentes do grupo.** Modelem juntos. Primeiro, porque é assim que você provavelmente vão modelar na vida

profissional. Segundo, porque o trabalho é um estudo para a prova e você deve saber fazer todos os diagramas na prova. Modelagem é como programação, só se aprende fazendo.

FAQ:

- **Porque o trabalho só vale 20% de nota?**
 - Porque é em grupo e com muitos componentes.
 - Porque é necessário tempo em sala de aula para acompanhar todos os trabalhos. Experiências anteriores mostraram que não é possível acompanhar um número maior de grupos.
 - Porque os alunos precisam demonstrar que aprenderam os conceitos da matéria na prova.
- **Vale a pena fazer?**
 - Dos alunos de períodos anteriores que responderam ao questionário de avaliação do curso: 67% declarou que o trabalho foi essencial para o aprendizado, 29% declarou que o trabalho ajudou muito no aprendizado. 0% responderam que o trabalho ajudou pouco ou nada no aprendizado. 2% não responderam.
- **Porque precisamos explicar tanto os diagramas?**
 - Não existe só uma modelagem correta, mas sim várias modelagens coerentes. É essencial que eu entenda o raciocínio de vocês para poder corrigir os diagramas. Quanto melhor vocês explicarem, mais chance de eu considerar consistente!