

INSTITUTO DE COMPUTAÇÃO – UNICAMP

MC857A - Projeto de Sistema de Informação

2024 - Semestre 1 - Prof. Jorge Stolfi

Aulas: 6^{as} 19:00–23:00 Prédio IC-3, Sala 304 (CC04)

Descrição preparada em 2024-03-04



Informações gerais

Objetivos da disciplina: Treinar o desenvolvimento eficiente e eficaz de sistemas complexos de informação, por uma equipe de dezenas de programadores trabalhando de forma semi-independente.

O termo *eficiente* implica, em particular, evitar a perda de tempo causada por incompatibilidades entre partes desenvolvidas por pessoas diferentes. O termo *eficaz* implica que o software pronto, além de cumprir seus objetivos, deve ser bem documentado, interna e externamente, robusto, e de fácil manutenção. Para atingir estes objetivos, o software deverá ser organizado em *módulos* que se comunicam através de *interfaces* simples e bem documentadas. *Programas de teste* deverão ser desenvolvidos para validar os principais módulos, e o sistema completo deverá ser validado em *conjuntos de teste*.

Formato do curso: A principal atividade da disciplina será a especificação, desenvolvimento e documentação de um sistema de software complexo, implementando um serviço em rede acessado via WWW, por todos os alunos em conjunto. O tipo de serviço será decidido no início do curso, e suas funcionalidades serão detalhadas e ajustadas no decorrer do mesmo.

O sistema será desenvolvido em `python3` para o sistema operacional GNU/Linux. (Justificativas para esta escolha serão apresentadas em classe.) O código-fonte do sistema será armazenado no repositório GitHub, projeto `JorgeStolfi/Projeto-MC857-2024-1`. Como a disciplina é eminentemente prática, as aulas (exceto possivelmente as provas) serão todas em sala de laboratório.

O desenvolvimento do projeto em si deverá ser feito **durante as aulas práticas**. **A presença física na sala de aula é obrigatória**. Haverá um único projeto, mas os alunos deverão trabalhar individualmente (com auxílio do professor, se necessário) nas tarefas individuais designadas no início de cada aula. **É importante a pontualidade no início da aula para permitir distribuição adequada das tarefas**.

Não haverá tarefas de programação específicas para casa (fora da aula). Porém, espera-se que cada aluno esteja familiarizado com a estrutura e interfaces principais de todo o software, e não apenas dos módulos em que trabalhou. **Este conhecimento será cobrado nas provas**.

Formato das aulas: No início de cada aula, cada aluno receberá uma cópia da versão corrente do software, na forma de um *branch* do repositório (um clone interno do projeto), e a especificação de sua tarefa. No decorrer da aula, deverá modificar essa cópia de modo a cumprir a tarefa que lhe foi atribuída. No final da aula, deverá enviar o código alterado para o GitHub (com `git commit` e `push`). A compatibilização e resolução de conflitos e a incorporação das mudanças de cada *branch* na versão mestre ficará a cargo do professor.

O envio das alterações ao GitHub deve ser efetuado no máximo até o final da aula, mesmo que a tarefa não tenha sido inteiramente cumprida. Naturalmente, quem terminar antes não precisa ficar até o fim da aula.

Antes de enviar seu código para o GitHub, cada aluno deverá incluir no seu *branch* um arquivo de **texto simples** chamado `relatorios/num.RA/2024-mm-dd.txt` com uma descrição sucinta de seu trabalho na aula (módulos desenvolvidos ou alterados, problemas encontrados, testes feitos, etc.).

Avaliação

Nota final: A nota final será baseada em duas provas escritas $P1$ e $P2$, com pesos 2 e 3, respectivamente; e uma nota de laboratório T . A média das provas P e a nota de laboratório T serão combinadas pela fórmula

$$M = (\max \{P, T\} + 4 \min \{P, T\})/5 \quad (1)$$

Para aprovação será necessário obter média final $M \geq 5,0$. Note que, para obter esta média, **é necessário (mas não suficiente!) ter média de provas P maior ou igual a 3,8, e também nota de laboratório T maior ou igual a 3,8.**

Provas As provas serão realizadas no horário normal da aula, nos dias 10 de maio ($P1$) e 28 de junho ($P2$). O exame final (E) será no dia 12 de julho.

As provas serão individuais, em classe, **sem** consulta. As questões versarão sobre a arquitetura geral do projeto (incluindo as principais interfaces, classes de objetos, e bases de dados), sobre os conceitos de computação relevantes para o mesmo, e, principalmente, **sobre as partes do mesmo que foram atribuídas especificamente a cada aluno.**

Exame: Haverá um exame escrito opcional E . O exame só poderá ser feito por quem tiver frequência suficiente nos laboratórios e tiver feito pelo menos uma das provas escritas. Caso seja feito, a média P acima será recalculada, tendo o exame peso igual à soma dos pesos das provas feitas. Ou seja, a média de provas P será calculada conforme a tabela abaixo:

Provas feitas	Média de provas
P_1, P_2	$P = (2P_1 + 3P_2)/5$
P_1	$P = (2P_1 + 3 \cdot 0)/5$
P_2	$P = (2 \cdot 0 + 3P_2)/5$
Nenhuma	$P = 0$
P_1, P_2, E	$P = (2P_1 + 3P_2 + 5E)/10$
P_1, E	$P = (P_1 + E)/2$
P_2, E	$P = (P_2 + E)/2$

Em qualquer caso, o aluno que comparecer a uma prova escrita (incluindo o exame final), e desistir de fazer (ou entregar a mesma em branco) depois de ver o enunciado, será considerado presente, e receberá nota zero nessa prova, sem direito a substituição.

Nota de laboratório: A nota de laboratório será atribuída com base na contribuição dada ao projeto. Essencialmente é $T = 10F/L$ onde F é o número de aulas de laboratório em que o aluno compareceu, e L é o número total de aulas de laboratório. Entretanto, cada aula poderá ser contada apenas em parte, ou mesmo excluída, se a tarefa designada não tiver sido cumprida satisfatoriamente.

Importante: Qualquer tentativa de fraude — nas provas ou nos trabalhos práticos, detectada na hora ou a posteriori — implicará na atribuição da nota zero *na disciplina*, sem direito a exame, *a todos os envolvidos*, sem prejuízo das demais sanções que possam ser tomadas.