Problema 2: Sistema de Atendimento Ambulatorial

Problema

Com a possível inauguração do Ambulatório da Universidade Estadual de Feira de Santana (UEFS), novas oportunidades se abrem para os estudantes e para a comunidade. Para os estudantes de Medicina, Enfermagem, Farmácia e outros, o Ambulatório traz a oportunidade de pôr em prática os conteúdos vistos ao longo das disciplinas teóricas e baseados na abordagem *PBL* (no caso de Medicina). Para a comunidade, significa mais uma unidade de atendimento à população da cidade.

Além de alguns recursos necessários à abertura do Ambulatório que devem ser concedidos pelo governo do Estado, a Reitoria da UEFS entrou em contato com a Assessoria Especial de Informática da UEFS (AEI) e informou sobre a necessidade de implantação de um sistema de atendimento aos pacientes. Percebendo o grande potencial de aprendizado envolvido no desenvolvimento deste sistema, o gerente de desenvolvimento da AEI entrou em contato e agendou uma reunião com o diretório acadêmico do curso de Engenharia de Computação (DAEComp).

Na reunião, o gerente de desenvolvimento da AEI explicou aos representantes do DAEComp como o sistema deve funcionar. Seguem alguns trechos importantes desta reunião:

...

Gerente: "*Em resumo, a Reitoria pediu que fizéssemos um sistema de atendimento por senha para os pacientes do Ambulatório. Os pacientes chegam à recepção e o recepcionista, por meio do nosso sistema, emite uma nova senha para este paciente*."

Representante do DAEComp: "Mas, e então? A pergunta inicial é: Que linguagem o senhor nos recomenda?"

Gerente:"Pois bem! É um sistema simples, ao mesmo tempo de grande fluxo...um sistema em Python é bastante viável! Podemos aproveitar de bons recursos do Python como os dicionários e as tuplas, por exemplo, e tornar nosso sistema leve e rápido."

...

Representante do DAEComp: "E quanto a estas senhas, que tal criarmos uma classe para elas?"

Gerente: "Não, não! Nada de classes e orientação a objetos! É como usar um canhão para matar uma barata! Podemos usar de estruturas simples, como eu havia dito antes! O simples uso de dicionários em conjunto com listas contendo os dados de cada paciente é suficiente e resolve nossos problemas. Por exemplo: cada senha pode estar associada a uma lista com os dados de um paciente em um ou dois dicionários diferentes para tipos diferentes de dados nas listas."

...

Representante do DAEComp: "É... acho que com isso podemos definir bem os requisitos do sistema!"

Gerente: "Ok! Só não se esqueçam da modularização do código e dos comentários! Precisamos de um código limpo, que permita o reuso de código no futuro e torne fácil a leitura e compreensão pelos próximos desenvolvedores que irão ampliar e atualizar este sistema!"

Após esta reunião, os representantes do DAEComp resolveram solicitar aos estudantes do módulo integrador de Algoritmos o desenvolvimento deste sistema. Para isto, eles formalizaram os requisitos coletados a seguir.

O sistema deve oferecer uma interface que permita ao funcionário/recepcionista do Ambulatório:

1. *Emitir nova senha*, identificando se o paciente é comum ou preferencial, seu horário de chegada (no formato *hora:minuto*) e qual o consultório de atendimento para esse paciente. Ainda deve ser informado o nome do paciente e este nome deve estar associado à sua senha.
2. *Chamar paciente para atendimento.* Nesta opção, o recepcionista libera a entrada do próximo paciente de determinado consultório (inicialmente, as especialidades oferecidas serão: Dermatologia, Endocrinologia e Ortopedia). Neste momento, o sistema deve determinar a próxima senha a ser chamada e exibir na tela a senha, o nome do paciente e o consultório que ele deve comparecer (ex.: Senha: c123 - Lucas Pereira - Consultório de Dermatologia). O recepcionista deve ainda, neste momento, informar ao sistema o horário de início da consulta deste paciente.
3. *Pular paciente*, caso haja desistência de algum paciente, este deve ser removido de todos os dados do sistema.
4. *Encerrar uma consulta* realizada em qualquer um dos consultórios do Ambulatório. Ao encerrar uma consulta, o funcionário/recepcionista deve ainda informar o horário de término da consulta do paciente.
5. *Exibir fila de espera*. Esta opção solicita a hora atual e mostra as filas de espera comum e prioritária de cada consultório como uma tabela: Senha, Nome do paciente, Em atendimento/Em espera, Tempo de espera na fila (para os pacientes em atendimento, esta opção deve exibir o tempo que aquele paciente esperou na fila até ser atendido).
6. *Exibir pacientes atendidos no dia* por consultório na forma de uma tabela: Nome do paciente, Tempo

de espera na fila, Tempo de Atendimento, Consultório, Médico do Consultório.

7. *Exibir tempo médio de espera dos pacientes*. Esta opção deve permitir calcular uma média do tempo de espera de todos os pacientes no ambulatório. Também deve exibir o tempo de espera médio dos pacientes comuns e dos pacientes das filas preferenciais no Ambulatório como um todo e por consultório.
8. *Exibir tempo médio de atendimento dos pacientes*. Esta opção deve permitir calcular uma média do tempo de atendimento de todos os pacientes no ambulatório. Também deve exibir o tempo de atendimento médio dos pacientes comuns e dos pacientes das filas preferenciais no Ambulatório como um todo e por consultório.

Além dos requisitos acima, os representantes do DAEComp ainda deixaram explicações adicionais sobre o funcionamento do sistema, descritas a seguir.

Ao chegar um novo paciente no ambulatório, o funcionário/recepcionista deve solicitar uma senha para este paciente. A senha de um paciente comum deve iniciar por letras específicas e ser complementada com o número do paciente indicando sua ordem de chegada na fila de espera. A senha ainda deve estar ligada ao tipo de especialidade necessitada pelo paciente, referente ao consultório. Por exemplo, o paciente "cd001" chegou para uma consulta de dermatologia antes do paciente "cd002", que também veio para o mesmo consultório. Após receber sua senha o paciente entra em uma fila de espera.

Há ainda uma fila diferenciada de espera para atendimento preferencial de idosos, portadores de necessidades especiais e mulheres grávidas. Este tipo de senha é identificado por letras específicas e sua ordem de chegada. Por exemplo, "pd012".

O sistema deve usar uma política de atendimento 50%-50%. Quer dizer, em média, deve ser atendido um paciente preferencial para cada paciente comum. Isto garante que os pacientes comuns que cheguem cedo não sejam prejudicados por um excesso de pacientes preferenciais que cheguem mais tarde. Obviamente, não havendo pacientes preferenciais em espera, os pacientes comuns podem ser atendido sem observar esta regra. E, vice-versa, não havendo pacientes comuns em espera, os pacientes preferenciais podem ser atendidos sem observar a regra. Mas, havendo os dois tipos de pacientes em espera, deve ser chamado primeiro um preferencial e, em seguida, devem ser alternados pacientes comuns e pacientes preferenciais.

Ao ser encerrada uma consulta, caso não haja nenhum paciente na fila preferencial, o paciente comum com a maior precedência de chegada deve ser chamado ao atendimento. Caso existam pacientes na fila de espera preferencial, deve-se respeitar a ordem de chegada dos pacientes considerando as duas filas, normal e preferencial, de cada consultório, e a política de atendimento 50%-50%. Por outro lado, caso o próximo paciente da lista de espera preferencial já tenha uma ordem de chegada anterior à um paciente comum, ele deve ser chamado para atendimento de qualquer modo, ignorando a regra 50%-50%.

Exemplo 1: Caso o próximo paciente, ce025, com maior precedência de ordem de chegada seja um paciente comum e o último paciente atendido tenha sido preferencial, este paciente ce025 pode ser atendido. Contudo, se o último paciente atendido tiver sido comum, ce025 terá que dar a vez ao preferencial pe030 que tem maior precedência dentre os preferenciais e, só depois, ser atendido.

Exemplo 2: Caso o próximo paciente, paciente po032, seja preferencial e tenha a maior precedência de ordem de chegada dentre as duas filas, ele será atendido, mesmo que o último paciente atendido tenha sido também preferencial. Após este paciente po032, novamente será analisado quem nas duas filas tem a maior precedência de ordem de chegada. Se for um paciente comum, ele poderá ser atendido, pois o último atendido, po032, era preferencial.

Exemplo 3: Caso o paciente co055 seja o paciente com maior precedência de ordem de chegada e a fila de pacientes preferenciais esteja vazia, ele será atendido normalmente e os próximos pacientes comuns serão atendidos um após o outro por ordem de chegada até que um novo paciente preferencial entre na fila de prioridade.

Após o atendimento, os pacientes devem ser incluídos em uma lista de pacientes atendidos.

Finalmente, cada consultório deve estar relacionado a um médico especialista da área. Dermatologia – Drª. Silvia Melo, Endocrinologia – Dr. Fernando Santos, Ortopedia – Drª. Maria do Carmo Silva.

Produto, Fluxograma e Relatório

Você deverá desenvolver o código fonte do sistema **adequadamente modularizado** em Python e um relatório final, no formato de artigo da SBC, conforme modelo e instruções disponibilizados no site do MI. O código deve se entregue até ao meio dia do dia **28/12/2019** e o relatório final até ao meio dia do dia **30/12/2019** (a entrega impressa do relatório final pode ser solicitada pelo tutor). O relatório final só será aceito mediante a entrega do código fonte.

O desempenho nas sessões tutoriais equivale a 30% da nota no Problema. O relatório, que inclui o fluxograma, equivale a 30% da nota e o código fonte (produto) equivale a 40%. Haverá penalidade de um ponto por descumprimento do prazo de entrega e um ponto por dia de atraso na entrega, até o máximo de cinco dias. Após este prazo, o trabalho não será mais aceito.

Tanto o código fonte quanto o relatório devem ser desenvolvidos individualmente. Deve constar no código fonte declaração de ausência de plágio.

Recursos para Aprendizagem

MANZANO, J. A. N. G.; OLIVEIRA, J. F. Algoritmos: Lógica para Desenvolvimento de Programação. São Paulo: Érica, 1996.

FORBELLONE, A. V. L., EBERSPACHER, H. F. Lógica de Programação: A Construção de Algoritmos e Estrutura de Dados. 2. ed. Makron Books, 2000.

WAZLAWICK, R. S. Introdução a Algoritmos e Programação com Python. Elsevier, 2018.

BORGES, L. E. Python para Desenvolvedores. Novatec, 2014.

SUMMERFIELD, M. Programação em Python 3. Elsevier / Altabooks, 2015.

DIERBACH, C. Introduction do Computer Science Using Python: A Computational Problem-Solving Focus. Wiley, 2012.

BEAZLEY, D.; JONES, B. K. Python Cookbook. O'Reilly, 2013.

BARRY, P. Use a Cabeça! Python. Elsevier / Alta Books, 2013.

Cronograma

Data	Sessão Tutorial
21/11/19	Apresentação do Problema 2
28/11/19	Sessão Tutorial - Problema 2
05/12/19	Sessão Tutorial - Problema 2
12/11/19	Sessão Tutorial - Problema 2
19/12/19	Sessão Tutorial - Problema 2
28/12/19	Entrega do Código-Fonte.
30/12/19	Entrega do Relatório.