

Universidade de São Paulo

Escola Politécnica da universidade de São Paulo, Departamento de Engenharia de
Computação e Sistemas Digitais

EPUSP - PCS3635 - Laboratório Digital 1
Turma 2 — Professor Edson Toshimi Midorikawa
Bancada A3



Projeto da Disciplina

Igor Pontes Tresolavy — NUSP 12553646
Thiago Antici Rodrigues de Souza — NUSP 12551411

2023/02/10
São Paulo

I. Possíveis temas e escopo

A. Ensinador de Braille

Um possível tema a ser explorado no projeto é um jogo que ensina Braille para pessoas que queiram se aprofundar no tema, ou que, devido a algum problema de saúde perderam a visão. Para tanto, o jogador receberia uma letra e precisaria traduzi-la para Braille utilizando uma série de botões assim como vistos no produto *FitLight*. Esse projeto possui o potencial de ser bem educativo, além de poder ajudar na reabilitação de algum paciente que tenha perdido a visão recentemente, ajudando na adaptação de sua nova condição.

Seria interessante implementar métricas que acompanham a evolução da capacidade da pessoa em sua tradução para Braille, como, tempo de resposta, acurácia das respostas, máximo de respostas acertadas consecutivamente, e outras métricas que ajudam no acompanhamento da evolução do paciente, ou interessado em Braille.

A *gamificação* do processo de aprendizado, também tem se mostrado muito vantajosa no nível de crescimento de aprendizado, possibilitando a aquisição de novas habilidades de maneira mais rápida, e sua retenção por mais tempo, tornando mais agradável o processo como um todo.

II. Recursos Disponíveis

Como caráter de pré requisito, precisamos utilizar da placa *DE0-CV*, e da *FPGA*, estudadas nas aulas prévias. Essas funcionariam como o a parte inteligente do circuito, responsáveis por fazer todo acompanhamento, incluindo analisar as métricas de performance do jogador e criar um relatório com tais métricas. Além disso, essa parte seria responsável por estabelecer as jogadas a serem feitas de maneira *pseudoaleatórias*, conseguindo, portanto, garantir um aprendizado completo do alfabeto em Braille.

Por conta da deficiência em questão, seria também necessário encontrar algum jeito de comunicar a tradução a ser feita de maneira não ótica. A maneira mais simples de se fazer isso é com *feedbacks* auditivos, para tanto, seria necessário a implementação de um auto-falante, que pudesse transmitir para o jogador, tanto a jogada a ser feita, quanto o *feedback* da jogada (acertou, errou, demorou muito tempo).

Para fazer a inserção da jogada, seria também interessante o uso de botões, configurados no padrão **3x2** presentes na matriz Braille, já criando a intuição necessária para se desenvolver a habilidade de leitura e escrita em tal linguagem.

Além disso, seria interessante ter uma ferramenta que integre com o sistema para apresentar as informações relevantes discutidas previamente, e, portanto conseguir realizar o acompanhamento de maneira granularizada, além de apontar mudanças no comportamento relevantes (p.e. redução de 50% no tempo de resposta, acurácia subiu 30%, etc).

III. Solução básica do sistema

Para realização do projeto, pode-se construir em cima do que já foi montado previamente na disciplina, com algumas pequenas alterações. Como todo circuito digital mais robusto, teremos uma *Unidade de controle*, que possui uma máquina de estados, responsável por gerar os *sinais de controle*, que controlam os componentes do sistema digital.

De maneira bem genérica, será preciso a amostragem da jogada, em que se comunica para o jogador a letra a ser jogada, e um estágio de coleta de jogada, em que se colhe a jogada que o jogador deve fazer e os dados pertinentes referente a essa jogada (como o tempo, se acertou, etc).

Também será necessário fazer a integração com os módulos do alto falante, para fazer a interface do *feedback* auditivo para o jogador, além de apontar a próxima letra necessária para se jogar, e a interface de botões, que registrará as jogadas feitas. O diagrama de alto nível do projeto pode ser visto na Figura 1.

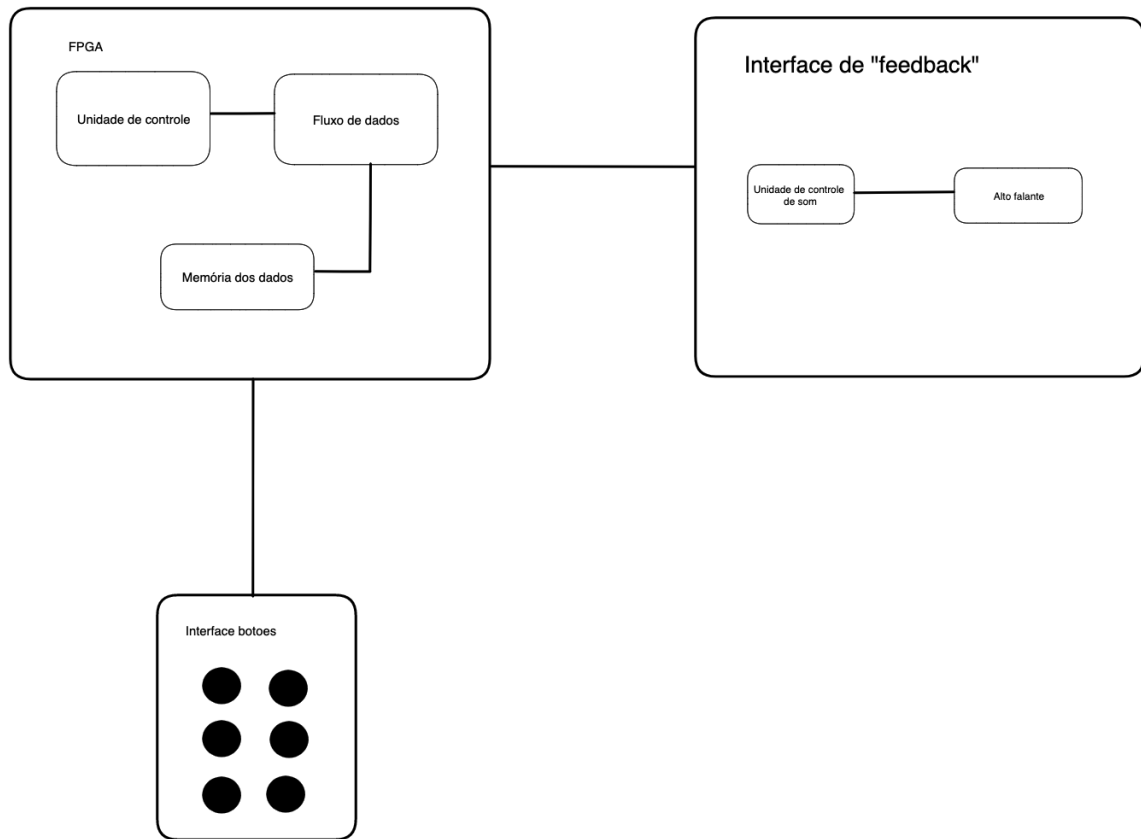


Figura 1: Diagrama de Alto Nível da implementação do projeto

IV. Requisitos Funcionais e Não Funcionais

A partir da definição do tema e seu escopo, levantou-se os requisitos funcionais e não funcionais especificados nas figuras a seguir, em formato de tabelas.

Código: BOTOES	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
Requisito: Botões de entrada da letra em Braille		
Descrição: Implementação dos seis botões que representam as protuberâncias do Braille, responsáveis por coordenar a letra no alfabeto Braille que o jogador insere no jogo.		
Prioridade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Estabilidade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Rationale: Deverão haver 6 botões organizados em uma matriz 3x2 que representa uma letra em braille. Idealmente, deve-se testar a matriz para cada letra do alfabeto.		
Requisitos associados:		

Figura 2: Tabela do requisito funcional *BOTOES*.

Código: BUZZER	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
Requisito: Implementação de um Buzzer para avisos sonoros		
Descrição: Buzzer que avisa erro e acerto de uma jogada (letra em Braille)		
Prioridade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Estabilidade:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Rationale: O Buzzer deve ser controlado diretamente pela FPGA ou pelo ESP32/ESP8266 e deve indicar se a letra jogada pelo jogador está correta ou incorreta, na jogada atual.		
Requisitos associados: ESP		

Figura 3: Tabela do requisito funcional *BUZZER*.

Código: ESP	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
Requisito: Uso de um microcontrolador para lógica de escrita		
Descrição: O microcontrolador se conectará à FPGA pelas suas GPIOs e a um outro dispositivo por Wi-Fi/Bluetooth, que enviará comandos correspondentes à letra adicional que deverá ser jogada pelo jogador na rodada seguinte.		
Prioridade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Estabilidade:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Rationale: Este requisito permite que a lógica de escrita seja operada por uma máquina ou um operador humano portando um dispositivo conectado ao microcontrolador (celular/computador) com interface para sua operação.		
Requisitos associados: BUZZER		

Figura 4: Tabela do requisito funcional *ESP*.

Código: AUDIO	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
Requisito: <i>feedback</i> auditivo da letra a ser jogada		
Descrição: Implementação de <i>feedback</i> auditivo por meio de um alto-falante que comunique ao jogador a nova letra a ser jogada adicionada na sequência.		
Prioridade:	<input type="checkbox"/> Alta	<input type="checkbox"/> Média <input checked="" type="checkbox"/> Baixa
Estabilidade:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Rationale: Este requisito permite que o jogador com deficiência visual seja comunicado da próxima jogada, não sendo necessário que um operador humano o faça.		
Requisitos associados:		

Figura 5: Tabela do requisito funcional *AUDIO*.

Código: ERROS	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
Requisito: Gerar dados sobre os erros cometidos durante o jogo		
Descrição: Apresentação da quantidade de erros total cometida durante as 16 rodadas do jogo.		
Prioridade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Estabilidade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Rationale: A apresentação da quantidade de erros total permite a quantificação do aprendizado do jogador.		
Requisitos associados:		

Figura 6: Tabela do requisito funcional *ERROS*.

Código: TEMPO	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
Requisito: Gerar dados sobre o tempo de o jogo		
Descrição: Apresentação do tempo médio para cada jogada durante as 16 rodadas do jogo.		
Prioridade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Estabilidade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Rationale: A apresentação do tempo médio por jogada permite a quantificação do aprendizado do jogador.		
Requisitos associados:		

Figura 7: Tabela do requisito funcional *TEMPO*.

Código: TIMEOUT	<input type="checkbox"/> Funcional	<input checked="" type="checkbox"/> Não Funcional
Requisito: Tempo máximo para cada jogada		
Descrição: Implementação de um temporizador que contabiliza um erro quando o tempo limite para uma jogada é atingido.		
Prioridade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Estabilidade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Rationale: A implementação de um tempo máximo adiciona um novo elemento à gamificação do aprendizado e serve como estímulo ao jogador.		
Requisitos associados: ERRO		

Figura 8: Tabela do requisito não funcional *TIMEOUT*.

Código: ORDEM	<input type="checkbox"/> Funcional	<input checked="" type="checkbox"/> Não Funcional
Requisito: Define a ordem das jogadas dos seis botões		
Descrição: Possibilidade do usuário pressionar os botões da jogada atual em qualquer ordem e individualmente.		
Prioridade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Estabilidade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Rationale: Dado que os botões na <i>protoboard</i> não simulam fielmente os blocos de letras em Braille, é necessário a possibilidade do acionamento dos botões em qualquer ordem e separadamente		
Requisitos associados:		

Figura 9: Tabela do requisito não funcional *ORDEM*.

Código: SOBREPOSICAO	<input type="checkbox"/> Funcional	<input checked="" type="checkbox"/> Não Funcional
Requisito: Define a possibilidade do aperto de botões simultaneamente		
Descrição: Possibilidade do usuário pressionar os botões da jogada atual simultaneamente.		
Prioridade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Estabilidade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Rationale: Caso o usuário deseje, deve ser possível que os botões sejam pressionados ao mesmo tempo e que a jogada seja contabilizada corretamente.		
Requisitos associados:		

Figura 10: Tabela do requisito não funcional *SOBREPOSICAO*.

Código: TEMPOMAX	<input type="checkbox"/> Funcional	<input checked="" type="checkbox"/> Não Funcional
Requisito: Define o tempo máximo antes do <i>timeout</i>		
Descrição: O tempo máximo antes do circuito acusar <i>timeout</i> na jogada deve ser de 30 segundos (aproximadamente 5 segundos para cada um dos seis botões).		
Prioridade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Estabilidade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Rationale: O usuário em aprendizado deve ter um intervalo de tempo máximo desafiador para cada jogada mas plausível de ser cumprido com sucesso.		
Requisitos associados: BOTOES, TIMEOUT		

Figura 11: Tabela do requisito não funcional *TEMPOMAX*.

Código: FREQUENCIA	<input type="checkbox"/> Funcional	<input checked="" type="checkbox"/> Não Funcional
Requisito: Define a frequência das notas emitidas pelo <i>buzzer</i> na sinalização dos erro e acertos		
Descrição: Deve-se diferenciar os sinais sonoros emitidos pelo <i>buzzer</i> nos eventos de erro e acerto de uma jogada.		
Prioridade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Estabilidade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
Rationale: Dado que o público alvo do dispositivo inclui pessoas com deficiências visuais, o circuito deve ser capaz de sinalizar o erro e acerto de uma jogada através de notas de frequências diferentes emitidas pelo <i>buzzer</i>		
Requisitos associados: BUZZER		

Figura 12: Tabela do requisito não funcional *FREQUENCIA*.

Algumas considerações acerca do funcionamento do circuito educativo podem ser apontadas. Mais especificamente, destacam-se, aqui, semelhanças e diferenças quanto ao seu funcionamento em relação ao circuito do Jogo da Memória. Como semelhanças, pode-se apontar que o jogo consistirá de 16 rodadas, sendo que, na primeira rodada, a sequência a ser inserida pelo jogador possui uma jogada—no contexto do projeto, uma jogada consiste na letra a ser inserida pelo jogador utilizando a matriz de botões (requisito *BOTOES*, da figura 2), que será comparada com a letra correta da jogada atual. A cada rodada, a sequência é incrementada em uma jogada, que será escrita pelo microcontrolador (requisito *ESP*, figura 4) ESP32 (ou ESP8266; não definiu-se, por enquanto, qual microcontrolador será utilizado), a partir de um operador humano (através de um aplicativo de computador e celular conectado ao microcontrolador por Bluetooth ou Wi-Fi). O microcontrolador conecta-se à FPGA pelas suas GPIOs (as conexões e arquitetura apresentam-se com mais detalhes na seção V). Finalmente, como no jogo da memória original, o circuito educativo implementa avisos sonoros (requisito *BUZZER*, figura 3) sobre os eventos do jogo (acerto e erro).

Quanto às diferenças, ressalta-se, aqui, o fato de que o jogo não acaba quando um erro é cometido. Ao inserir uma jogada incorreta, o circuito contabiliza o erro cometido (requisito *ERROS*, figura 6) e comunica o evento sonoramente ao jogador por meio de um *buzzer*. Na ocorrência de Timeout (requisito *TIMEOUT*, figura 8), o circuito também contabiliza erro e passa para a próxima jogada, não encerrando o jogo.

V. Revisão da Arquitetura da Solução

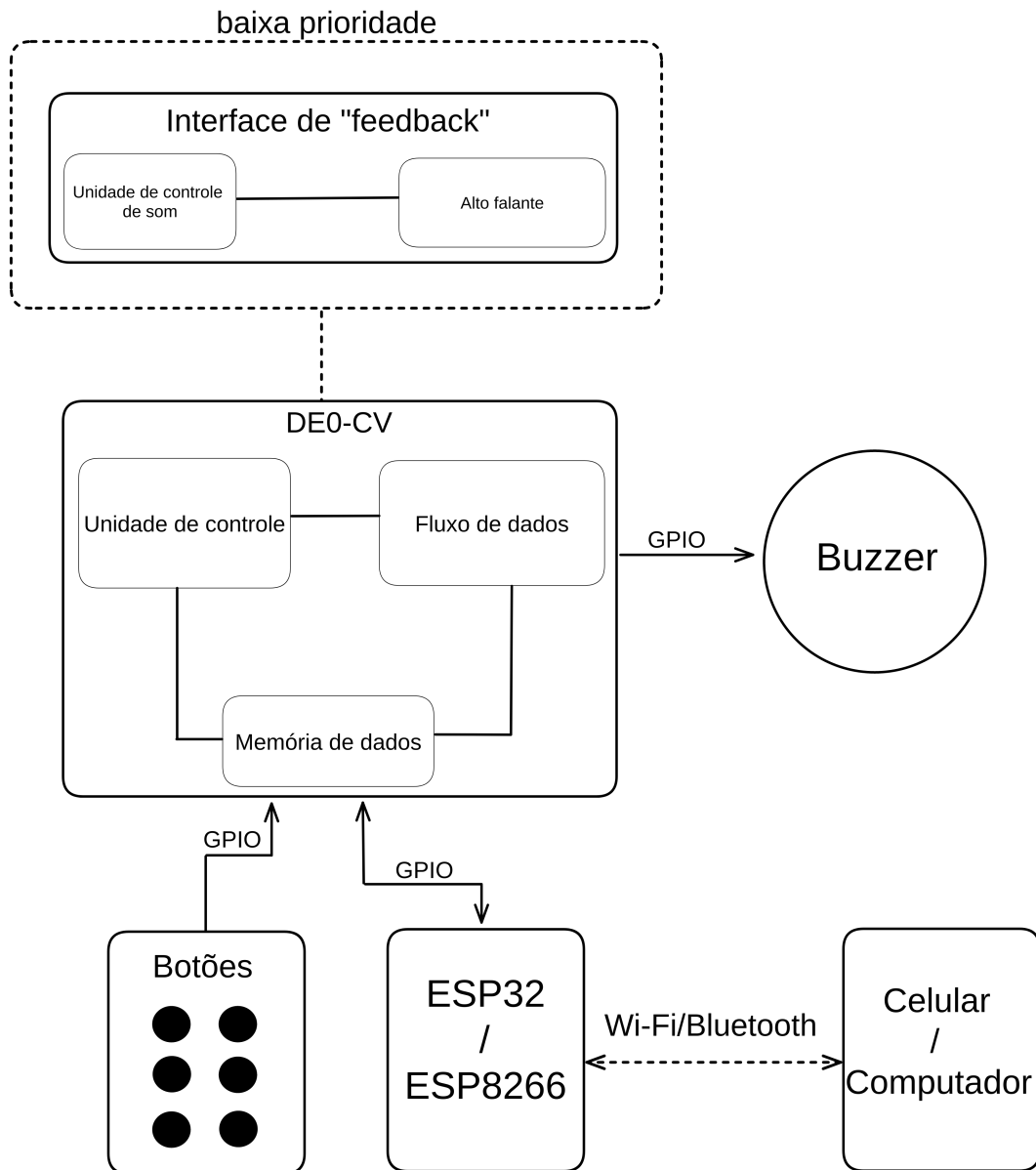


Figura 13: Diagrama de Alto Nível da implementação do projeto

A figura 13 apresenta uma versão revisada da arquitetura do sistema da figura 1. Destaca-se a adição do microcontrolador ESP (referente ao requisito funcional *ESP*), que conecta-se a um celular ou computador por Bluetooth ou Wi-Fi (ainda não especificados no projeto). Também nota-se a adição do *buzzer* responsável pelos avisos sonoros do jogo. Ademais, rotulou-se a interface de *feedback* como *baixa prioridade* por motivos explicitados em VID. Finalmente, também definiu-se o meio e direção pela qual esses blocos se comunicam. No caso, a grande maioria se comunicam pelas GPIOs da DE0-CV, com exceção da comunicação de via dupla entre o celular e o microcontrolador. A direção entrada/saída está indicada por setas.

VI. Cronograma das atividades

A. Semana 1:

Pretende-se implementar os requisitos *BOTOES*, *TIMEOUT*, *ORDEM*, *SOBREPOSICAO* e *TEMPO-MAX*. O requisito *TIMEOUT* já foi implementado em experiências anteriores e será somente modificado nesse projeto a fim de cumprir com os requisitos não funcionais *TEMPOMAX*.

Os requisitos não funcionais *ORDEM* e *SOBREPOSICAO* podem ser implementados concomitantemente, dado que ambos serão definidos pelas decisões de projeto tomadas durante a fase de implementação do requisito funcional *BOTOES*.

B. Semana 2:

Segue-se com a implementação dos requisitos *BUZZER* e *FREQUENCIA*. Para essa etapa do projeto, deve-se decidir se o controle sonoro do *buzzer* será realizado através do microcontrolador escolhido (ESP32 ou ESP8266), ou diretamente pelo circuito sintetizado em FPGA. Por fim, deve-se definir quais frequências do som emitido pelo *buzzer* associam-se ao erro e ao acerto de uma jogada.

C. Semana 3:

Implementação do requisito funcional *ESP*. Aqui, deve-se implementar uma interface humano-máquina capaz de permitir que um operador se comunique com o microcontrolador escolhido (ESP32 ou ESP8266) por Bluetooth ou Wi-Fi, possibilitando o envio da letra a ser escrita na nova jogada de cada rodada.

D. Semana 4:

Por fim, na última semana, prossegue-se com a implementação dos requisitos *ERROS* e *TEMPO*. A quantidade máxima de erros e o tempo médio por jogada devem ser apresentadas em *displays* de sete segmentos da placa DE0-CV, do laboratório. Esses dados, apesar de não serem acessíveis ao jogador com deficiência visual, podem ser utilizado para geração de métricas por parte do aplicador do jogo.

Caso haja tempo de projeto, prossegue-se para a implementação do requisito funcional de menor prioridade: *AUDIO*. Definiu-se esse requisito como de baixa prioridade pois, dado o escopo da disciplina, acredita-se que a complexidade do requisito excede as restrições de tempo impostas no projeto.