

**Pontifícia Universidade Católica de
Campinas - PUC-Campinas**

Documento de Visão Projeto NotaDez

Componente Projeto Integrador 2
Curso de Engenharia de Software - 2025

Autores: Prof. Me. Mateus Dias, Profa. Dra. Renata Arantes e
Prof. Dr. Luã Muriana

Campinas - 2025

1. 0 que é este documento

Este documento trata do escopo e limites da proposta de projeto chamada NotaDez que deverá ser implementada no decorrer da disciplina de projeto integrador 2.

1.1. Autores

Prof. Me. Mateus Dias, Profa. Dra. Renata Arantes e Prof. Dr. Luã Muriana.

1.2. Quem deve ler esse documento de visão

Dedicado para os estudantes que estão cursando neste semestre a disciplina de Projeto Integrador 2 do curso de Engenharia de Software.

1.3. Encontrei um erro ou problema neste documento. O que devo fazer?

Contatar seus professores orientadores da disciplina de Projeto Integrador 2 para retirar as suas dúvidas e fazer ajustes caso sejam coerentes e necessários.

1.4. Direitos autorais

Este documento está protegido nos termos da lei de direitos autorais de 9.610/98. Portanto, este documento não deve ser encaminhado de maneira digital ou impressa para outras pessoas externas à disciplina e que não sejam os próprios alunos de Projeto Integrador 2, matriculados em 2025 no curso de Engenharia de Software.

Nenhuma ideia, solução, parte ou textos integrais contidos neste documento poderão ser reaproveitados, por quem quer que seja, fora do contexto estritamente acadêmico da disciplina Projeto Integrador 2 do curso de Engenharia de Software, para o ano de 2025.

Portanto, qualquer pessoa que tenha recebido este documento sem autorização, apague-o imediatamente e informe os autores relatando como este documento chegou até você.

Porém, se caso você deseje aproveitar ideias aqui contidas com a finalidade comercial (mesmo que sem custo), soluções propostas e requisitos, escreva um e-

-mail formal aos autores, enviando sua proposta de uso para obter ou não uma autorização. Só serão analisados pedidos realizados por meio de um profissional jurídico ou escritório especialista em propriedade intelectual, com procuração devidamente concedida. Pedidos feitos sem o envolvimento de um profissional da área jurídica ou escritório especialista, serão automaticamente recusados e sequer analisados.

Endereços de e-mail dos autores:

mateus.dias@puc-campinas.edu.br

renata.arantes@puc-campinas.edu.br

lua.marcelo@puc-campinas.edu.br

2. 0 Projeto NotaDez

Na atualidade, docentes do ensino superior precisam de uma ferramenta particular e única para gerenciar as notas de seus estudantes, muito além de sistemas acadêmicos institucionais pois é propriedade intelectual do docente ter documentado as notas de seus estudantes ao longo dos seus anos de atuação. Essa não é uma informação apenas institucional, mas de propriedade compartilhada entre: professor, estudante e instituição.

Costumeiramente, os docentes utilizam o Excel ou editor de planilhas para fazer controle pessoal de notas e, de fato são softwares excelentes e consagrados. No entanto, muitas vezes não há integração desses aplicativos com outras ferramentas institucionais, pois não é tão trivial realizar esse trabalho por parte das instituições e, além disso, por ser uma ferramenta mais ampla e tratar de diversos casos de uso, não atende especificamente esse nicho de problema: o controle de notas.

Então, pensando nisso, ao utilizar NotaDez, o docente cadastrará uma ou mais instituições onde ele trabalha com um poucas informações tais como nome, as disciplina que leciona, também poucas informações (código, nome e apelido) e turmas (código, nome e apelido). Importará para o sistema os estudantes ou poderá cadastrá-los um a um vinculados a uma turma. Na sequência natural de uso, poderá apontar notas em diversas atividades, provas e possibilitará o cálculo de nota final da disciplina de maneira automática.

A aplicação funcionará 100% Web e terá a capacidade de exportar e importar dados para facilitar o trabalho do professor. Além disso, poderá ser hospedada na nuvem ou servidor dedicado para que o professor possa, de qualquer lugar, utilizá-la.

3. Requisitos do NotaDez

3.1. Autenticação

O docente que desejar usar o NotaDez, deve criar uma conta de acesso usando um endereço de e-mail pessoal. Para a criação da conta, serão exigidos os dados:

Nome, e-mail, telefone celular e senha. O sistema deverá também ter um recurso de recuperação de senha por meio de e-mail “esqueci a minha senha”.

Não existirá qualquer acesso anônimo para este projeto ou modo visitante. Ao acessar o sistema, a primeira página a ser exibida é a página de autenticação e, a partir dali, o professor visitante cria uma conta ou se for cadastrado, prosseguirá para sua utilização.

Não há qualquer necessidade de hotsite ou páginas explicativas. Esse projeto é totalmente um software web, não se trata de um portal informativo ou qualquer outra taxonomia de site web que necessite de textos longos, vídeos e campanhas de marketing.

3.2. Gerenciar instituições, disciplinas e turmas

Após o cadastro, quando o docente acessa o sistema pela primeira vez, deve informar pelo menos uma instituição onde trabalha e um curso que leciona. Por exemplo: PUC-Campinas e Engenharia de Software. Em seguida, poderá criar as diversas disciplinas e turmas. Entende-se por turma, um conjunto de estudantes que frequentam uma disciplina em um determinado dia, horário e local.

Então, após o docente informar a instituição e um curso, poderá prosseguir com o cadastro de disciplinas informando: Nome, sigla, código e período do curso (1 semestre, 2 semestre como desejar) e criar quantas turmas que ele quiser, por exemplo: Turma 1, Turma 2, etc associadas às disciplinas.

Quando o docente quiser excluir uma turma que já possua notas lançadas, o sistema enviará um e-mail e o docente deverá clicar em um link para efetivar a exclusão. Portanto, entende-se como adequado em termos de segurança essa confirmação, para garantir que o docente não exclua acidentalmente uma turma totalmente completa, com muitas notas lançadas. A exclusão de uma turma, será uma operação irrevogável.

Da mesma maneira, o docente não poderá, por exemplo, excluir uma instituição que tenha disciplinas e turmas. Deverá fazer a exclusão de cada uma coisas na ordem natural de dependências: Excluir primeiro turmas, após a disciplina e se não houver nenhuma outra disciplina, poderá excluir a instituição. No entanto, somente

a exclusão de turmas é crítica e solicitará a confirmação por e-mail (confirmar a operação de exclusão clicando em um link de confirmação da exclusão).

3.3. Cadastro de alunos ou importação de arquivo .CSV/JSON de alunos

Sempre que o docente acessar uma determinada turma de uma disciplina, poderá cadastrar quantos estudantes desejar (incluir, editar alunos existentes ou removê-los um a um ou por múltipla seleção).

O docente poderá importar estudantes para uma turma a partir de arquivos dos tipos: CSV¹ e JSON². Ao visualizar o CSV em uma ferramenta como o Excel, os dados apresentados serão semelhantes a uma tabela:

| Matrícula | Nome |
|------------------|-------------------|
| 11111 | Abel Antimônio |
| 11112 | Bianca Nióbio |
| 11113 | Carla Polônio |
| 11114 | Carlos Zinco |
| 11115 | Leonardo Plutônio |
| 11116 | Matheus Basalto |
| ... | ... |

Nota: sempre que o docente importar estudantes (e poderá fazer isso a qualquer momento), a ferramenta utilizará somente as DUAS PRIMEIRAS COLUNAS do arquivo CSV, considerando a primeira como sendo o identificador do aluno e a segunda como o nome completo do estudante. Esse será o padrão do sistema. Qualquer outra coluna deverá ser DESCONSIDERADA no momento de importação de alunos.

Para importar alunos a partir de um arquivo JSON, o arquivo deverá ser um array de subdocumentos. Cada subdocumento representa um estudante e, obrigatoriamente, serão considerados os campos matrícula, id, código ou RA como identificador e o nome ou fullName ou completeName ou simplesmente name como nome do estudante.

Além disso, caso já existam estudantes naquela turma e o docente faça uma nova importação de dados, o sistema verificará se existem duplicatas pelo identificador do estudante. Não poderão existir dois estudantes com o mesmo identificador (RA,

¹Comma-separated version

²JSON - Java Script Object Notation

Matrícula, ID - chame-o como quiser). Caso exista esse conflito, considerar sempre os dados no NotaDez como fonte da verdade e não sobrescrevê-los.

3.4. Criar componente de nota

Qualquer exercício, atividade, tarefa ou prova que o docente aplicou para uma turma e deseje registrar no sistema, se chamará componente de nota.

Afinal, o componente de nota é entendido como uma parte da composição de um cálculo maior que define a nota final para cada estudante em uma determinada disciplina.

Para exemplificar, suponha que a nota final da disciplina Cálculo seja a média de três notas, sendo cada uma das provas o valor de 0 a 10.

$$\text{Nota Final} = \frac{P1 + P2 + P3}{3}$$

Logo, o docente deverá informar (cadastrar) três componentes de nota para aquela disciplina (para aplicar em todas as turmas de maneira idêntica). Cada componente será sempre uma nota numérica que vai de 0.00 até 10.00 e poderá aceitar: valores como: 0.55, 0.99, 1.81, 10.00, 9.50. Respeitando sempre duas casas de precisão.

Exemplo: Supondo que o docente importe uma turma com 5 alunos e crie já os 3 componentes PARA A DISCIPLINA: P1 (prova 1), P2 (prova 2) e P3 (prova 3). Ao acessar a turma no sistema, após cadastrar os componentes de nota na disciplina, aparecerá uma tabela parecida:

| Matrícula | Nome | P1 | P2 | P3 |
|-----------|-------------------|-----|-----|-----|
| 11111 | Abel Antimônio | - | - | - |
| 11112 | Bianca Nióbio | - | - | - |
| 11113 | Carla Polônio | - | - | - |
| 11114 | Carlos Zinco | - | - | - |
| 11115 | Leonardo Plutônio | - | - | - |
| 11116 | Matheus Basalto | - | - | - |
| ... | ... | ... | ... | ... |

A tabela sempre abrirá em modo de visualização para que o docente não edite sem querer uma determinada nota.

No cadastro do componente de nota será necessário informar: Nome, Sigla (usada na fórmula) e Descrição. Exemplo: Nome: Prova 1, Sigla para fórmula: P1 e na descrição: Prova teórica sobre conteúdo do primeiro módulo.

3.5. Apontar notas dos componentes

Na interface de quadro de notas daquela turma, o docente poderá apontar somente notas de um único componente por vez, por exemplo, P1 ou P2. No entanto poderá selecionar uma opção chamada Modo de Edição Completa, que permitirá o docente mexer a vontade em qualquer uma das notas dos alunos, não importando qual componente seja.

A equipe de projeto deverá criar um mecanismo de interface capaz de controlar a edição das notas individualmente por componente ou no modo edição total.

O modo edição total, geralmente é utilizado quando o docente deseja, por exemplo, editar diversas notas que são de componentes distintos com cuidado para não alterar as notas indesejadas.

Nesta versão do projeto, não controlaremos recursos de desfazer ou refazer notas já modificadas.

3.6. Painel de auditoria

A cada nota lançada pela primeira vez ou modificada, o sistema deverá salvar e mostrar em uma caixa de auditoria de notas as mensagens de cada lançamento ou alteração em detalhes.

Ao lado da tabela de notas existirá um painel de auditoria que funcionará como um registro de LOG e exibirá mensagens em texto de feedback ao usuário. Exemplo: O docente alterou a nota do João de 5 para 5.5. Deverá aparecer uma mensagem: dd/mm/yyyy HH:MM:ss - (Aluno João Silva) - Nota de 5.0 para 5.5 modificada e salva. Onde dd/mm/yyyy HH:MM:ss significa a data e horário atual que o servidor registrou a operação. Só poderá aparecer mensagens no painel de auditoria das notas que foram alteradas e CONFIRMADAS pelo backend da aplicação.

Conforme o docente vai mexendo nas notas, o painel de auditoria vai mostrando sempre um "LOG" de operações para que ele tenha certeza do que está acontecendo

e confiança que as notas foram mesmo salvas. Essas mensagens precisam ser salvas no banco de dados e, sempre que carregar aquela tela de notas para trabalhar, as mensagens todas devem ser ordenadas por data/hora decrescente.

Esse painel pode ser ocultado ou exibido pelo docente, quando quiser. Não ficará fixo na interface. Mas, a auditoria é obrigatória, nenhum docente poderá desabilitá-la. O fato de esconder o painel, não desabilita o recurso.

3.7. Cálculo de notas finais³

Ao acessar a disciplina (que contém 1 ou mais turmas), o sistema verificará se existe um ou mais componentes de nota cadastrados. Se sim, vai liberar um campo texto chamado Nota Final. Neste campo, o docente informará a expressão matemática para cálculo da nota final a partir dos componentes informados. Por exemplo neste caso:

$$(P1 + P2 + P3) / 3$$

O sistema deverá ser capaz de interpretar a fórmula informada pelo professor e antes de salvá-la verificar se todos os componentes cadastrados estão utilizados na fórmula.

É um desafio para a equipe resolver essa parte do sistema. Em outras palavras, o docente salvar um “texto” que representa uma fórmula e o sistema fazer o cálculo automaticamente de acordo com o critério especificado.

Quando o professor informar a fórmula de nota final e voltar para a tela de lançamento de notas da turma, sempre visualizará a coluna de cálculo da nota final sem poder editar os dados que estão apresentados pois se trata de um cálculo automático e que o docente não poderá mexer.

3.8. Coluna Notas Finais Ajustadas

Quando o docente informar a fórmula de cálculo para uma determinada disciplina, existirá uma opção: [Sim/Não] acrescentar coluna de notas finais ajustadas. Se o docente selecionar (sim ou marcar) essa opção na interface. Sempre serão exibidas

³É importante destacar que os professores de web ou projeto integrador 2 não vão tirar dúvidas ou auxiliar as equipes a resolver este desafio ou implementar este requisito. Se trata de um desafio.

duas colunas de notas finais: Notas Finais (Cálculo automático) e ao lado Notas Finais Ajustadas (que o docente poderá mexer).

De maneira geral, nas Universidades que utilizam notas de 0 a 10 e não conceitos de A a E, existe a questão do arredondamento. As notas são fracionadas de 0.5 em 0.5. Então a nota final de um aluno será por exemplo: 5.0 ou 5.5 e não 5.8. Caso o cálculo dê 5.8, o sistema detecta que 0.8 está mais próximo de 1.0 do que de 0.5 e faz o arredondamento da nota para cima: 6.0 (mas esse arredondamento só pode acontecer na coluna final ajustada). Caso o aluno tenha tirado 5.4, 0.4 está mais próximo de 0.5 e então o arredondamento será 5.5, enquanto 5.1 está mais próximo de 0 e o arredondamento seria para baixo. Quando existem casos em que a nota automática seja, por exemplo, 5.25, 0.25 é exatamente metade entre 0.0 e 0.5, neste caso o arredondamento é para baixo: 5.0. Mas, quando o aluno por exemplo tirou 5.30, 0.3 está acima de 0.25 o que torna mais próximo de 0.5. Então a nota é ajustada para 5.5.

Ao acessar a primeira vez as notas dessa coluna, elas estarão já calculadas de meio em meio respeitando as regras do parágrafo anterior.

A coluna de nota final ajustada sempre terá precisão de um único número, fracionados em meio. Por exemplo: 10.0, 8.5, 6.5, 7.0.

Não serão aceitas notas nesta coluna do tipo: 6.4, 8.3, 7.8 etc.

Por quê essa coluna é necessária? Justamente para tratar casos excepcionais. Supondo que um aluno tirou exatamente 4.5 sem arredondamento, entendendo que a nota de corte da instituição seja, por exemplo, 5.0. Esse estudante seria reprovado. No entanto, o docente poderia conceder 0.5 ponto a mais por conta de algum trabalho extra de acordo com sua própria autonomia e decisão de docente.

Então, para não perder a referência, a coluna de cálculo automática sempre estará ao lado, para que o docente avalie bem os arredondamentos e possa deslocar a nota para cima ou para baixo, respeitando os intervalos de meio em meio ponto de precisão.

4. Requisitos de ambiente

Os requisitos obrigatórios de ambiente de desenvolvimento para qualquer equipe do projeto integrador são:

Backend: Node.js última versão LTS, com linguagem TypeScript.

Frontend: HTML5, CSS3 e pode se usar o Bootstrap de maneira opcional, mas recomendada;

Banco de dados relacional, desde que seja algum desses SGBDs: MySQL, PostgreSQL ou Oracle.

IDEs (Ambientes Integrados de Desenvolvimento): Microsoft Visual Studio Code ou JetBrains WebStorm.

Git e GitHub. Não serão aceitos outros serviços de repositórios remotos como Bitbucket ou outro sistema de controle de versões como mercurial.

5. Regras elementares do Projeto Integrador

5.1. Nome do repositório

No GitHub a equipe deve criar um repositório, exatamente no seguinte padrão: ES-PI2-ANO-TURMA-NUMERO-GRUPO. Exemplo de como nomear: Considere que a sua equipe curse PI-2 na turma 2 e seu grupo seja de número 03. O nome do repositório deve ser **obrigatoriamente**: ES-PI2-2025-T2-G03. Caso não siga exatamente a regra e também em tudo maiúsculo, o grupo não será advertido, será simplesmente penalizado em 1.0 ponto final na nota de todos os integrantes por tamanha falta de observação.

5.2. Armazenamento de código e contribuições dos membros

O código-fonte será armazenado em um repositório no GitHub, obrigatoriamente. Não será considerada outra ferramenta por mais privilegiada que seja.

O grupo deve estudar e aplicar o conceito de branches. Exemplo: se o grupo está trabalhando nesse momento na “autenticação” do projeto, deverá usar uma branch

(ramificação) exclusiva para isso e, ao terminar, integrar o código na branch master ou main e fazê-lo funcionar.

Commits de desenvolvimento devem ser feitos sempre considerando branches de desenvolvimento. Esse conceito é explicado no livro ProGit.

Todos os membros serão avaliados em relação à produção de código e autoria de seus artefatos pelo GitHub. Portanto, se você não sabe utilizar o Git é imprescindível que aprenda.

Se algum membro da equipe não estiver entre os membros do projeto no GitHub desde o início do projeto, será automaticamente reprovado sem direito a qualquer reclamação.

5.3. Release de entrega final

Para a entrega final, sua equipe deve criar uma TAG no repositório git chamada 1.0.0-final. Caso não exista a TAG de release, o grupo será inteiro penalizado com 1.0 a menos na nota final de TODOS os integrantes.

5.4. Arquivo README.md

Ao visitar o repositório da equipe, os professores precisam se deparar com uma descrição do que o seu projeto é, quem é a sua equipe, como colocar seu projeto para funcionar para testes e todos os detalhes necessários a respeito de como poder pegar o código que está lá no GitHub e poder implantá-lo num ambiente de testes. Todos esses cenários devem ser descritos num arquivo chamado README.md e que deve ser colocado na RAIZ do repositório.

Caso a equipe não tenha feito o README.md até a data da entrega final, todos os integrantes terão desconto de 1.0 ponto na nota final.

5.5. Comentários nos códigos

Todo código deverá conter comentários explicativos para facilitar a compreensão das partes pelos avaliadores da banca. Além disso, cada artefato (arquivo) seja de

qualquer linguagem: HTML, CSS, TypeScript, JavaScript etc, deverá ter mencionado no topo do arquivo quem foi o autor exclusivo daquele artefato.

Não serão aceitos projetos em que todos os arquivos sejam elaborados por todos os autores. A individualidade de produção deve ser respeitada.

Cada arquivo sem comentários ou sem identificação do autor, todos os integrantes sofrerão desconto de 1.0 ponto na nota final.

5.6. Apontamento de esforço

As tarefas deverão ser organizadas usando obrigatoriamente a ferramenta GitHub Projects. Aponte TODAS as tarefas e horas utilizadas no projeto, de maneira real. Isso faz parte do aprendizado. É um componente curricular obrigatório.

Se os orientadores perceberem que as equipes não estejam apontando corretamente o esforço ou que estejam mascarando o trabalho ou até mesmo colocando cards apenas para dizer que estão usando a ferramenta, mas sem qualquer intenção real de aprendizado ou uso efetivo, toda a equipe poderá ser desclassificada antes de ir para a banca, pelo professor da disciplina.

Motivo: displicência no apontamento de horas.

5.7. Participação nas reuniões de orientação

O time deverá ter reuniões periódicas para acompanhamento do progresso com o professor(a) do PI2. Caso algum membro não compareça OU não apresente nada do que fez durante o período passado até a reunião, sofrerá descontos em sua nota individual de comprometimento. Não basta estar na reunião, precisa apresentar o que fez.

Estar na reunião e também não participar, levará também a desconto de pontos de acordo com cada situação específica, conforme o orientador decidir.

5.8. Convite de participação no repositório

Atenção para a ordem de convites de participação no repositório:

Inicialmente, a equipe deverá convidar apenas seus integrantes e também o(a) docente orientador(a) para entrar como membro no GitHub do projeto. Portanto, se você e a sua equipe é orientada apenas pelo professor Luã, convide-o para participar do GitHub já no início do projeto. Caso seja a Renata, convide-a ao iniciar seu projeto efetivamente de acordo com as orientações deles.

Quando se aproximarem as datas das bancas, os professores orientadores de projeto integrador, informarão cada equipe quem (pessoas) farão parte da banca avaliadora. Quando as datas estiverem próximas, os orientadores informarão o e-mail de cada membro da banca, seja ele interno e/ou externo para que a equipe convide-os para acessar o repositório do projeto. Isso deverá ser feito **somente e, tão somente, quando a equipe entregar o projeto em sua versão final e o orientador informar para quem e qual endereço de e-mail os convites deverão ser enviados.**

Convites feitos no início do semestre ou no decorrer para professores diversos supondo que participarão da banca, não serão aceitos, mesmo que a equipe saiba antecipadamente. O motivo disso é bastante simples: os convites para os repositórios no GitHub tendem a expirar, então devem ser realizados em data apropriada, quando o orientador autorizar.

5.9. Banca avaliadora

Todos os grupos e seus membros apresentarão o projeto para uma banca avaliadora numa data e horários específicos que será divulgada em momento oportuno.

A dinâmica da apresentação será conduzida pela banca. A banca poderá informar como deseja que o grupo apresente, de forma livre e espontânea. Quem conduz o que quer assistir na apresentação é a banca, não os alunos. Portanto se a banca desejar começar diretamente pela arguição poderá. A equipe terá um tempo total máximo de 20 minutos, contando o tempo de preparação (setup).

Portanto, as equipes devem estar preparadas para: fazer uma apresentação do aplicativo funcionando, ter os códigos prontamente funcionando e sem **qualquer** necessidade de adaptação ou correção em momento de banca.

Caso a equipe tenha algum problema com equipamentos, falta de partes, códigos que pararam de funcionar ou qualquer outro imprevisto, terá apenas 3 minutos (não

adicionais) para resolver. Caso a equipe não consiga resolver o problema neste tempo, a banca interromperá a apresentação e desclassificará todos os membros por falta de organização e preparo antecipado. É inadmissível em qualquer banca de qualquer instituição, candidatos despreparados e sem plano de contingência.

Recomenda-se realizar um preparo sério e antecipado. Nenhuma equipe deve contar com a sorte. Deve ter o controle exato do que está acontecendo. No dia da apresentação para a banca, nada deve ser conduzido de forma relaxada ou despreparada por parte das equipes.