

Desenvolvimento de um Sistema de Gestão Escolar com o uso da Linguagem *Dart* com *Framework Flutter*

Matheus Maião Franklin

Faculdade de Tecnologia de Garça - matheus-maiao@hotmail.com

Ronaldo Aparecido Samuel Filho

Faculdade de Tecnologia de Garça - ronaldo.samuel@fatec.sp.gov.br

Resumo

A gestão escolar envolve muitas informações e serviços diretamente associados à eficiência dos processos escolares. Quando a gestão é automatizada por um sistema computacional, o tempo gasto em trabalhos operacionais pode ser reduzido significativamente. Esse trabalho possui como objetivo mostrar através de pesquisas bibliográficas os resultados do desenvolvimento de um aplicativo para *smartphones* com a Linguagem *Dart* e *framework Flutter*. Mesmo que relativamente simples, o aplicativo pode ser de muito útil na gerência de uma instituição, otimizando e organizando os trabalhos de docentes e dos funcionários administrativos, permitindo uma maior organização das informações que a instituição gera durante o ano letivo, como registros das aulas, notas, faltas e desempenho do aluno no final de cada bimestre. Para a conclusão, o sistema proposto poderá ser avaliado, por meio de uma pesquisa qualitativa, mensurando, assim, quais foram os resultados da implementação do aplicativo e qual foi a sua contribuição como ferramenta facilitadora dos processos gerenciais.

Palavras-chave: Software de gestão escolar. Gestão escolar. Framework Flutter. Linguagem Dart.

Development of a School Management System Using the Dart Language with Framework Flutter

Abstract

School management involves a lot of information and services directly associated with the efficiency of school processes. When a computer system automated the management, the operational work time used reduced significantly. This work aims to show through bibliographic research the results of the development of an application for smartphones with Dart Language and Flutter framework. Although relatively simple, the application can be utterly useful in the management of an institution, optimizing and organizing the work of teachers and administrative staff, allowing a better organization of the information that the institution generates during the school year, such as class records, grades, absences and student performance at the end of each quarter. In conclusion, the proposed system can be evaluated, through qualitative research, measuring, thus, what were the results of the application implementation and what was its contribution as a tool that facilitates the management processes.

Keywords: School management software. School management. Flutter framework. Dart language.

1 Introdução

Indubitavelmente, nos dias atuais, a disseminação da tecnologia no meio social se encontra de forma extensiva, pois as mudanças têm sido cada vez mais aceleradas e envolvendo grandes e crescentes volumes de dados. A revolução informacional tem impactos profundos em

todos os setores da sociedade, sobretudo, pela amplitude e velocidade com que dados são transmitidos e conexões estabelecidas.

O cenário evolutivo da tecnologia da informação ao longo da história vem acompanhado das mudanças tecnológicas que definiram as chamadas Revoluções. Desde a Revolução agrícola, passando pela Revolução Industrial a Revolução da Informação tiveram profundos impactos na sociedade gerando rearranjos das relações entre as pessoas (RABANAL; RAMIREZ, 2010, p. 2).

A gestão escolar pode encontrar vários obstáculos quando há falhas na organização das informações necessárias para sua execução. Em uma escola, todas as ações devem ser registradas de modo a criar uma rotina procedimental conhecida por todos e que seja passível de conferência pela equipe gestora, caso contrário prazos podem ser perdidos, metas não alcançadas e, conseqüentemente, todo o processo de ensino prejudicado. De acordo com Corrêa (2003 p. 21), “na média, qualquer trabalhador, em qualquer ramo de atividade, depende cerca cinco, de suas oito horas diárias de trabalho, com informação” (*apud* RABANAL; RAMIREZ, 2010 p. 2).

Seria onerosa a realização manual de serviços que envolvem muitas informações, que podem se tornar muito repetitivos, por exemplo, o cadastro de notas de toda a escola. Por esta razão soluções por meios de aplicativos e softwares computacionais podem trazer bons resultados no que tange à automatização desses processos.

Muitas instituições como universidades fazem o uso de um sistema de gestão acadêmica onde são armazenadas informações relevantes para os estudantes, como suas notas, presença, conteúdo explorado em cada aula, documentação e outras funcionalidades. Não apenas essas instituições, mas também sistemas utilizados pelo governo, como a Secretaria Escolar Digital (SED):

A Secretaria Escolar Digital é uma plataforma on-line com vários módulos. Criada para centralizar, agilizar e facilitar todas as operações que envolvem a gestão diária da administração escolar, professores, alunos e seus responsáveis, de forma rápida, segura e eficiente promovendo a inclusão digital. (SED, 2019, p. 1).

A pressuposição de barreiras na criação de sistemas como: altos custos, formação de profissionais capacitados entre outras são citadas por Teixeira e Brandão (2003), contudo, podem ser contornadas com criatividade e pesquisa. Isso porque muitas ferramentas computacionais são softwares livres, ou seja, não há custo com licenças e há muitos profissionais “*free-lancer*” que disputam por vezes o mesmo nicho de mercado, com isso, os custos com a mão de obra qualificada acaba por ser reduzido.

Para a construção de um sistema de grande porte e para sua implementação e uso corretos, de modo a gerenciar de maneira eficiente a instituição, o desenvolvedor e analista deve ter profundos conhecimentos de programação e de engenharia de software, a fim de melhor definir os requisitos e reduzir possíveis vulnerabilidades que poderiam existir com uma modelagem mal organizada.

O sistema a ser desenvolvido auxiliará na automatização de rotinas realizadas pelo corpo docente, como lançamento de notas e faltas, a fim de otimizar o tempo gasto, permitindo um melhor aproveitamento do docente com outras atividades pedagógicas, como a preparação de materiais didáticos para as referidas aulas.

A construção do *software* busca aprimorar a gestão do tempo e de mecanismos de registros de dados mediante a organização de algoritmos, viabilizando a melhoria dos processos de busca de informações, compartilhamento e acesso por meio de interface simples e funcional.

Com essa automatização, os processos gerenciais ganharão melhor desempenho, contribuindo de uma forma geral até mesmo para a padronização procedimental dentro da instituição, onde os indivíduos envolvidos na construção de documentos referentes a esses atributos serão direcionados pela padronização fornecida pelo próprio *software*.

Esse projeto foi proposto com a intenção de melhorar e apoiar a gestão escolar pelo uso das novas tecnologias; para isso, procura abranger os responsáveis justamente por essa tarefa, conforme os conceitos de gestão democrática escolar de Luckesi (2007) e os de tecnologia escolar abrangido por Teixeira e Brandão (2003).

Diante das demandas da instituição observada durante a realização desse projeto, é proposto um sistema que atenda às necessidades específicas dessa unidade escolar, buscando auxiliar o processo de gestão. O software desenvolvido é voltado à realidade escolar e suas particularidades e, a partir dele, busca-se fomentar discussões para o uso desse tipo de tecnologia e analisar os resultados alcançados.

Os objetivos específicos desta pesquisa estão relacionados à disponibilização de recursos por meio de um aplicativo que permita: a automação de planilhas (contabilização de notas de avaliações diagnósticas e da própria instituição e ausências), antes criadas pelo corpo docente e sem padronização de formato ou armazenamento; fortalecimento das interações entre os professores, alunos e familiares por meio de um *chat* para acompanhamento das atividades escolares e esclarecimento de dúvidas, sem que seja necessário o uso de aplicativos externos como *Whatsapp*, *Messenger* entre outros, ação que contribuirá também para a formalização dos registros e posterior verificação, caso necessário.

2 Desenvolvimento

2.1 *Back-End* em uma Abordagem Geral

O *Back-End* tem por objetivo tratar de toda a parte de processos internos, ou seja, tudo o que é executado de forma transparente ao usuário. Praticamente nesta parte do sistema, não há ou existe muito pouca interatividade com o usuário. São linhas de código que tratam em grande parte da manutenção dos bancos de dados, muitas vezes instalados em servidores na chamada nuvem cibernética.

Um *Back-End* bem construído contribui positivamente para um aplicativo e ou software com mais responsividade, atualizações rápidas, conciso e eficiente. Em jogos, por exemplo, quanto mais otimizado for a escrita de código de seu *Back-End* melhor desempenho ele terá, incluindo a execução gráfica. O setor de jogos é bastante cobrado pelos usuários e, para atender o esperado, o desenvolvedor deve retirar partes supostamente desnecessárias de código, as quais fariam o software ter um desempenho inferior, já que com mais linhas para o compilador efetuar a tradução binária, a execução seria mais lenta.

2.1.1 *Back-End* na Aplicação Prática do Projeto

O *Dart* é a ferramenta de suporte escolhida para este projeto, pois é uma linguagem que possui diversas funcionalidades, sem contar sua flexibilidade, possuindo integração com *Frameworks* e serviços tais como: *Flutter* e *Firebase*, de maneira que esses dois últimos serão também utilizados na construção do programa.

O *Dart* foi criado pela Google em 2011 com o objetivo de substituir o JavaScript no desenvolvimento de aplicações web. Entretanto por utilizar variáveis de tipos específicos podemos considerar o *Dart* como uma linguagem fortemente tipada e multiparadigma, que se consiste em uma linguagem que permite que o mesmo objetivo seja alcançado de diferentes formas como diz Jungthon e Goular (2009):

Um paradigma é o que determina o ponto de vista da realidade e como se atua sobre ela, os quais são classificados quanto ao seu conceito de base [...] uma linguagem de programação pode combinar dois ou mais paradigmas para potencializar as análises e soluções. (JUNGTHON; GOULAR, 2009, p.1)

O uso da linguagem *Dart* foi escolhida por conta de sua integração com o *Framework Flutter*, assim como o *JavaScript* pode ser usado como uma linguagem *server side* com a implementação do *node*, a linguagem *Dart*, possui essa semelhança.

O *Framework Flutter* usa a *thread* da linguagem *Dart* para fazer conexão com a *thread* nativa do aparelho na qual estiver sendo executada, dessa forma o aplicativo terá sua um desempenho já que mesmo que não foi desenvolvido em *Java* ou *Kotlin* no caso do *Android* ou *Swift* para o *IOS* o desempenho será semelhante, mesmo possuindo o mesmo código para ambas plataformas.

2.2 *Front-End* em uma Abordagem Geral

A parte frontal, a programação de mais alto nível, chamada de *Front-End* é onde se concentra o contexto que fará contato com o usuário, ou seja, a interface do sistema. O desenvolvimento de um bom *Front-End* e sua importância, está no fato de ser o primeiro contato com o usuário, será a “primeira impressão com o usuário”.

Uma interface bem construída permite que o usuário de modo prático tenha acesso a todos os recursos e funcionalidades do aplicativo ou sistema, evitando que funcionalidades importantes fiquem escondidas em meio a menus e abas desnecessárias.

Certos sistemas que são voltados para usuários mais avançados por vezes são bonitos mas todos os seus recursos não são explícitos para usuários mais leigos, pode-se citar o sistema operacional *Windows* desenvolvido pela *Microsoft*, neste exemplo, a maioria dos usuários podem ter acessos aos recursos principais do sistema, entretanto nem sempre possuem conhecimento para utilizar serviços como configurações avançadas de rede e ou gerenciamentos de dispositivos por exemplo. Por ser um sistema muito abrangente ainda é insuficiente quanto a sua interface para a grande maioria dos usuários.

2.2.1 *Front-End* na Aplicação Prática do Projeto

A função do *Framework* é resolver alguns problemas recorrentes com uma abordagem genérica, permitindo ao desenvolvedor focar seus esforços na resolução do problema em si, e não ficar reescrevendo *software*. Por isso foi escolhida o *framework Flutter* para este projeto, o *Flutter* é um *framework* de código aberto sobre a *BSD Licence* e multiplataforma, a linguagem base do *Flutter* é o *Dart*.

2.3 *Back End* e *Front End* – questões de projeto

Para o *Back End* foi definido o uso do *Firebase*, por ser uma plataforma de desenvolvimento de aplicações abrangente, tendo várias funções, por exemplo: controle de acesso, banco de dados e ainda algumas análises que poderão ser feitas no aplicativo em questão. O *Firebase* foi criado pela Google, em setembro de 2011.

Até a presente data, estão prontas as telas de protótipos, suas funcionalidades e como elas irão se interligar, para isto, foi usado um tema minimalista, para que dessa forma o usuário possa interagir com facilidade no uso com uma interface simples e de fácil manuseio. A interface possui cores mais escuras pois assim os ícones ficam com mais foco e em destaques.

A ferramenta utilizada para a prototipação da interface foi o *Justinmind Prototyper*, que se consiste numa ferramenta com *templates* prontos para a elaboração de interfaces comuns em sistemas para *smartphone*. No processo de desenvolvimento será explorado o uso de ferramentas como: *Android Studio* para a integralização da interface com o Back-End e essa ferramenta permite a utilização do *Android Virtual Device* (virtualização de um sistema *Android* que permite emular as aplicações ainda não finalizadas), *Visual Studio Code* para a escrita do código *Back-End*.

Um dos objetivos iniciais para desenvolver o aplicativo de gestão escolar é o uso de *smartphones*. Assim, as tecnologias definidas são: O *framework Flutter* para a criação do *Front-End*, *Firebase* para a armazenagem de dados *Back-End* do *software* junto com a linguagem *Dart*.

Tais definições se consolidam, pois o *Flutter* integrado com o *Firebase* apresenta uma boa estrutura e facilidades na sincronização. Além disso, o *Firebase* possui boa capacidade para armazenar os dados, e podendo fazer aplicações tanto para *IOS* quanto *Android* sem a necessidade de recodificação.

O desenvolvimento contém a exposição ordenada e pormenorizada do assunto pesquisado. Divide-se em seções e subseções que variam em função da abordagem do tema e do método. Inicia-se com uma revisão teórica acerca do assunto pesquisado, contempla os procedimentos metodológicos utilizados na pesquisa e apresenta os resultados seguidos das discussões.

2.4 Desenvolvimento da *Interface*

Muitos sistemas de gestão escolar possuem a mesma forma de trabalhar: um aplicativo no qual o usuário tenha acesso às informações cadastradas e um portal onde os administradores inserem tais informações, em neste projeto, tais funcionalidades estão em um único aplicativo, dependendo de qual grau de permissão que o usuário tenha, opções serão disponibilizadas. Assim, por exemplo, um diretor pode cadastrar todos os tipos de informações, desde funcionários e alunos, mas, não notas e faltas, pois estas são funções apenas de professores e, estes por sua vez, podem cadastrar alunos notas e faltas, mas não outros professores por exemplo. Aos alunos, apenas consultas.

Alguns protótipos de interface cada um com seu propósito:

- *Login*: nesta interface um usuário já cadastrado poderá ter acesso ao seu sistema.
- *Home*: nesta interface o usuário poderá visualizar os últimos eventos tanto da instituição como notas de atualizações, avisos etc.
- *Notas*: visualização da nota de cada aluno.
- *Faltas*: visualização das faltas de cada aluno
- *Horários*: separados por dia, o aluno poderá acessar os dados da hora da aula, matéria e professor que irá lecionar.
- *Mais*: A opção “Mais” irá aparecer apenas para gestores (permitindo-se cadastrar professores, alunos e aviso) e professores (podendo-se cadastrar notas faltas e alunos).

A figura 1 consiste no protótipo da tela inicial/*login* sendo a tela que o usuário irá visualizar assim que o programa for iniciado.

Figura 1. Tela Inicial/*Login*

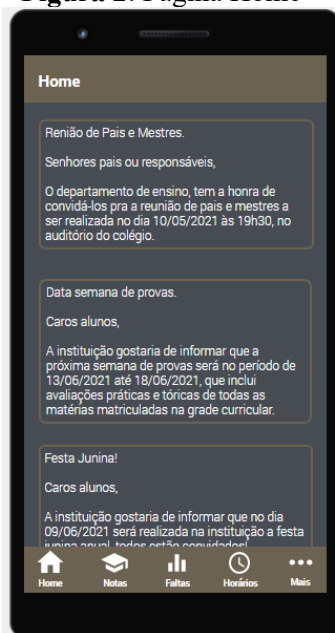


Fonte: Elaborada pelos autores (2020).

A parte superior do projeto possui seu logotipo junto com sua sigla e, abaixo encontra-se dois campos, um para inserir o usuário e o outro para inserir sua senha, em seguida há dois botões um para efetuar o *login* que em seguida irá direcioná-lo para página *home* do aplicativo, considerando a hipótese que o usuário erre uma das informações, ele será notificado por meio de um aviso na interface.

A figura 2 apresenta as funcionalidades do programa: a tela *Home* que se consiste na tela principal com mensagens, anotações, e alguns lembretes para o aluno.

Figura 2. Página Home



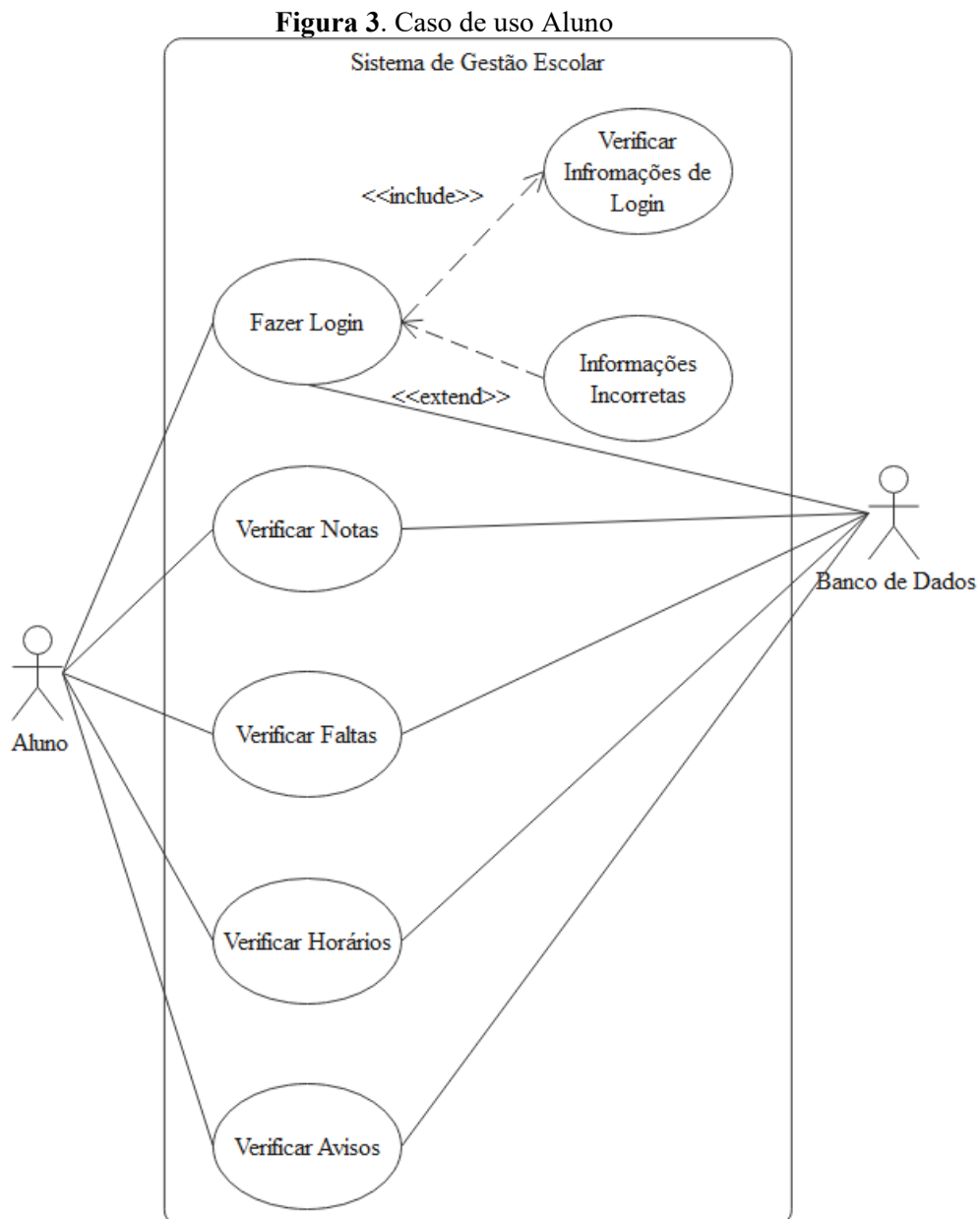
Fonte: Elaborada pelos autores (2020).

No rodapé do aplicativo possui uma barra que é possível interação para navegação no aplicativo como: notas, faltas e horários. Na tela de notas, como o próprio nome a define, é a tela que irá apresentar as notas que o aluno possui no semestre e, na tela de faltas irá ser apresentado as faltas que o aluno já sofreu e, por fim, na tela de horários será apresentado ao usuário os horários de aula. Outra funcionalidade, localizada no botão “Mais”, permite que gestores (diretores, coordenadores), possam cadastrar alunos e os professores as informações de seus alunos, como por exemplo informações como as demonstradas no exemplo 2.

2.5 Desenvolvimento dos casos de uso

O diagrama de caso de uso permite que um ideia mesmo que complexa, seja explanada de forma simples e direta, por meio de relações entre ações que um dado sistema ou software.

Podemos verificar na figura 3 a relação entre usuário (aluno) e banco de dados (*Fire Base*).



Fonte: Elaborada pelos autores (2020).

Ao efetuar o *login*, uma verificação de informações será realizada, após essa atividade considerando a hipótese de informações incorretas uma mensagem será exibida na interface do sistema, caso suas credenciais estejam corretas, a possibilidade de consulta das informações desejadas será disponibilizada, como por exemplo a verificação de suas notas, faltas etc. Porém não terá permissão para cadastramento de informações diretamente no banco de dados.

No exemplo 4 será apresentado o relacionamento do professor ou gestor com o banco de dados.



Fonte: Elaborada pelos autores (2020).

O gestor herdará todas ações do professor para tal poder fazer os registros, quando o professor efetuar *login*, será feita a mesma verificação realizada para o aluno garantindo a confiabilidade das credenciais do sistema, evitando que um indivíduo tenha permissões que não lhe pertencem. Como professor ele poderá registrar notas e faltas dos alunos, como o gestor

registrará os alunos em suas respectivas disciplinas, cadastrar professores dados e registrar os horários de aula.

2.6 Etapas futuras

Após a fase de conclusão da implementação será necessária uma variedade de testes, como por exemplo o próprio uso do software, verificação de trechos desnecessários e releitura de código, visando uma melhora no desempenho do sistema com o objetivo de encontrar falhas de codificação e possíveis vulnerabilidades no projeto, também serão solicitados que pessoas utilizem o sistema de modo que possa fomentar de discutir a respeito da *interface*, melhorando-a e garantindo um ambiente agradável e de fácil utilização. Assim, após tais testes, serão feitas alterações, dependendo das necessidades, partindo de análises e resultados obtidos pelos orientadores e resultados encontrados na etapa de teste do projeto, logo após essas correções o aplicativo estará pronto para utilização.

O aplicativo poderá também evoluir de acordo com os *feedbacks* dos usuários. Os quais poderão comentar sobre os pontos de vista de cada um deles, podendo também contribuir para aumentar a experiência do usuário, onde sugestões dos usuários serão usadas para melhorar as funcionalidades do aplicativo.

3 Conclusão

A pesquisa científica apresentada se dá por meio de pesquisa exploratória de forma que o projeto ainda se encontra em fase de desenvolvimento. Assim, como resultados esperados vislumbra-se uma melhor gestão nos fastidiosos processos desempenhados pela instituição. A intenção é que facilite certas ações, otimizando tempo e melhorando a organização de documentos essenciais em uma instituição de ensino.

As *interfaces* serão projetadas de forma a serem o mais amigáveis possível, porém, a integração do sistema talvez não seja tão adaptável a todos logo de início por passar pelo período de habituação em relação a algo novo, entretanto, a medida que a frequência da usabilidade aumenta, poder-se-á melhorar algumas *interfaces* a fim de atender a demanda dos usuários em questão.

Uma das características do aplicativo proposto no projeto é que, em um mesmo sistema, todos poderão interagir e isso será feito por adaptações no sistema de *login/senha*, pois prioridades e funcionalidades serão atribuídas de acordo com o *status* de cada usuário. Em outras palavras, um professor terá acesso a opções diferentes dos alunos, assim como um diretor. Dessa forma, tendo como expectativa que o uso do sistema seja mais prático e otimizado, permitindo assim que uma interface intuitiva com total proveito do potencial da aplicação.

Além disso, dados confidenciais serão protegidos pelo banco de dados já que o Firebase conta com um sistema de criptografia próprio e embarcado em suas funções, área de *chat* do sistema e outros. Com isso, espera-se um ambiente seguro para a área de informações do sistema.

REFERÊNCIAS

FREIRE, Fernanda Maria Pereira; PRADO, M. E. B. B. Projeto Pedagógico: Pano de fundo para escolha de um software educacional. **O computador na sociedade do conhecimento**, v. 1, 1999.

JUNGTHON, Gustavo; GOULART, Cristian Machado. **Paradigmas de Programação**. Monografia (Monografia)—Faculdade de Informática de Taquara, Rio Grande do Sul, v. 57, 2009.

LUCKESI, Carlos Cipriano. Gestão Democrática da escola, ética e sala de aula. **ABC educativo**, n. 64, 2007.

RABANAL, Sissiliana B. del R. Vilchez de; RAMIREZ, Pedro Aldo Rabanal. **Alinhamento Estratégico de Negócios e TI em Ambiente de Hipercompetição**. In: SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA, 7., 2010, Rezende: Seget, 2010. v. 7, p. 2

SED - Secretaria Escolar Digital. **Sobre a Secretaria**. Disponível em: <https://sed.educacao.sp.gov.br/sobre-a-secretaria>. Acesso em 27 nov. 2019.

SENA, Moisés. **O que é educação? Brasil Escola**, Disponível em: <https://meuartigo.brasilecola.uol.com.br/educacao/o-que-educacao.htm>. Acesso em: 02 dez. 2019.

SOMMERVILLE, I. **Engenharia de Software**. São Paulo: Pearson-Addison Wesley, 2003.

TEIXEIRA, Adriano Canabarro; BRANDÃO, Edemilson Jorge Ramos. Software educacional: o difícil começo. **RENOTE**, v. 1, n. 1, 2003.