

**UNIVERSIDADE PAULISTA – UNIP**

Jorge Willian dos Santos - D44830-7

Leandro da Silva Oliveira - D32091-2

Wesley Aparecido Nunes de Assis – D399IB1

Kaique Curan Ribeiro - N1079E-1

**UNIINFO**

**SOROCABA**

**2018**



Jorge Willian dos Santos - D44830-7  
Leandro da Silva Oliveira - D32091-2  
Wesley Aparecido Nunes de Assis – D399IB1  
Kaique Curan Ribeiro - N1079E-1

## **UNIINFO**

Trabalho de conclusão de  
semestre para obtenção do título de  
graduação em (ADS) apresentado a  
universidade paulista – UNIP

Orientador: Prof. Richardson Luz

**SOROCABA**

**2018**

## FICHA CATALOGRÁFICA

CIP - Catalogação na Publicação

dos Santos, Jorge Willian  
Uniinfo / Jorge Willian dos Santos. - 2018.  
0192 f.

Trabalho de Conclusão de Curso (Graduação) apresentado ao Instituto de Ciência Exatas e Tecnologia da Universidade Paulista, Sorocaba, 2018.

Área de Concentração: Tecnologia da Informação.

Orientador: Prof. Esp. Richardson Luz.

Coorientadores: Prof. Esp. Marcos Cruz, Prof. Esp. Randall Gasparini, Prof. Esp. Reverdan Sparinger.

1. Aplicativo. 2. Software. 3. Uniinfo. 4. Consulta. 5. Localização. I. Luz, Richardson (orientador). II. Cruz, Marcos (coorientador). III. Gasparini, Randall (coorientador). IV. Sparinger, Reverdan (coorientador). V. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da Universidade Paulista com os dados fornecidos pelo(a) autor(a).

Jorge Willian dos Santos - D44830-7  
Leandro da Silva Oliveira - D32091-2  
Wesley Aparecido Nunes de Assis – D399IB1  
Kaique Curan Ribeiro - N1079E-1

## UNIINFO

Trabalho de conclusão de  
semestre para obtenção do título de  
graduação em (ADS) apresentado à  
universidade paulista – UNIP

APROVADO EM:

### BANCA EXAMINADORA

Prof. Robinson Luz  
Universidade Paulista - UNIP

\_\_\_\_\_/\_\_\_\_/\_\_\_\_

Prof. Marcelo Bellodi  
Universidade Paulista - UNIP

\_\_\_\_\_/\_\_\_\_/\_\_\_\_

Prof. Eduardo Simi  
Universidade Paulista - UNIP

\_\_\_\_\_/\_\_\_\_/\_\_\_\_



*“Todos os homens têm, por natureza, desejo de conhecer”.*

(Aristóteles.)





## **RESUMO**

Software auxiliar de consulta multiplataformas Uniinfo, tem como principal objetivo fornecer informações referentes à Universidade Paulista UNIP, auxiliando estudantes, professores e visitantes na busca de locais, eventos, entre outras informações. Um grande diferencial do projeto é a praticidade de uso e versatilidade contidas no produto. Caso deseje conhecer os eventos decorrentes, horários de cursos, localizações e demais informações, haverá abas específicas com as devidas informações.

O Banco de dados se encontra em um servidor em nuvem para consulta e recebimento de dados advindo de funcionários da UNIP, em função de regular a informação contida no aplicativo. O Uniinfo é um sistema portátil, gratuito e multiplataformas, que auxiliará não apenas estudantes na busca da legibilidade do funcionamento da UNIP.

**Palavras-chave :** Software auxiliar , Uniinfo, multiplataformas.



## **ABSTRACT**

Uniinfo's multiplatform auxiliary consultation software, has as main objective to provide information regarding the UNIP Sorocaba SP university, helping students, teachers and visitors in the search of places, events, among other information. A great differential of the project is the practicality of use and versatility contained in the product. In case you want to know the events, courses schedules, locations and other information, there will be specific tabs with the appropriate knowledge.

The database is located on a cloud server for consultation and receipt of data from UNIP employees, in order to regulate the information contained in the application; another site was also developed for more questions regarding the application. Uniinfo is a free, cross-platform, portable system that will not only help students in seeking the readability of UNIP's operation

**Palavras-chave :** Auxiliary software, Uniinfo, multiplatform.



## Sumário

1.	APLICATIVO .....	17
1.1	Uniinfo .....	17
1.1.1	TypeScript.....	17
1.2	Abas .....	18
1.2.1	Eventos .....	18
1.2.2	Cursos .....	18
1.2.3	Locais .....	19
1.2.4	Informações .....	19
2	WEB SERVICE.....	21
2.1	Banco de Dados.....	22
3	HEURISTICAS .....	25
3.1	Visibilidade do estado do sistema .....	25
3.2	Correspondência entre o sistema e o mundo real .....	26
3.3	Liberdade de controle fácil ao usuário.....	26
3.4	Consistência e padrões .....	27
3.5	Prevenções de erros .....	28
3.6	Reconhecimento em vez de memorização .....	29
3.7	Flexibilidade e eficiência de uso .....	30
3.8	Estética e design minimalista .....	31
3.9	Ajuda ao usuário a recuperar-se de erros .....	32
3.10	Ajuda e documentação.....	33
4	UML .....	35
4.1	Caso de Uso.....	35
4.1.1	Diagrama de Caso de Uso.....	37
4.1.2	Diagrama de Classe .....	37
4.1.3	Diagrama de sequência .....	38
5	ANÁLISE DE RISCOS.....	39
5.1	Análise de ameaças .....	39
5.2	Avaliação dos riscos.....	39



5.3	Tratamento dos riscos .....	40
6	PADRÕES DE QUALIDADE - ISO/IEC 9126 (NBR 13596) .....	41
6.1	Introdução ISO/IEC 9126 .....	41
6.2	Modelo de qualidade de software.....	41
6.2.1	Processo de desenvolvimento .....	41
6.2.2	Produto .....	41
6.3	Qualidade em uso .....	42
6.3.1	Funcionalidade (satisfaz as necessidades?).....	42
6.3.1.1	Adequação.....	42
6.3.1.2	Acurácia.....	42
6.3.1.3	Interoperabilidade .....	42
6.3.1.4	Segurança.....	43
6.3.2	Confiabilidade (é imune a falhas?).....	43
6.3.2.1	Maturidade .....	43
6.3.2.2	Tolerância a falhas.....	43
6.3.2.3	Recuperabilidade .....	44
6.3.3	Usabilidade (é fácil de usar?).....	44
6.3.3.1	Inteligibilidade .....	44
6.3.3.2	Apreensibilidade .....	44
6.3.3.3	Operacionalidade.....	44
6.3.4	Eficiência (é rápido e enxuto?).....	45
6.3.4.1	Comportamento em relação ao tempo.....	45
6.3.4.2	Utilização de recursos.....	45
6.3.5	Manutenibilidade (é fácil de modificar?).....	45
6.3.5.1	Modificabilidade .....	45
6.3.5.2	Estabilidade .....	45
6.3.5.3	Testabilidade.....	46





6.3.6	Portabilidade (é fácil de migrar para outro ambiente?) .....	46
6.3.6.1	Adaptabilidade .....	46
6.3.6.2	Capacidade para ser instalado .....	46
6.3.6.3	Coexistência .....	47
6.3.6.4	Capacidade para substituir .....	47
6.3.6.5	Analisabilidade .....	47
6.4	Conclusão .....	47
7	CONCLUSÃO .....	49
	REFERÊNCIAS BIBLIOGRÁFICAS .....	51
	APÊNDICE .....	53
	APÊNDICE A - ANÁLISE DE RISCOS .....	55
	APÊNDICE B – DIAGRAMAS UML .....	57
	APÊNDICE C – TELAS DO UNIINFO .....	59
	APÊNDICE D - CÓDIGO FONTE APLICAÇÃO WEB .....	73
	APÊNDICE E – CÓDIGO FONTE APLICATIVO MOBILE .....	127



## 1. APLICATIVO

### 1.1 Uniinfo

O aplicativo Uniinfo foi desenvolvido com a principal ideia de ser um guia informativo sobre os locais da instituição, como a localização de salas, blocos, andares, bibliotecas, secretaria etc. Além disso, é oferecido suporte sobre eventos e datas como, provas, palestras, data de entrega de trabalhos, entre outros serviços oferecidos por essa plataforma de consulta.

Imagem 1 - Logo Uniinfo



Fonte – Uniinfo 2018

O principal conceito formado sobre as funcionalidades do software são, facilidade de uso, praticidade e velocidade, ou seja, ele foi desenvolvido para auxiliar o cliente na busca de informações referentes à Unip, de forma simples e objetiva. Para sua estruturação foi desenvolvida nas linguagens HTML, CSS e TypeScript.

#### 1.1.1 TypeScript

Criado pela Microsoft, o TypeScript está provando ser uma escolha comum entre os desenvolvedores ASP.NET, que não é uma nova linguagem, mas sim de um superset (ou superconjunto) do JavaScript. Tem de recursos que melhor suportam o uso da Programação Orientada a Objetos, que tem como base quatro princípios fundamentais, encapsulamento, herança, abstração e polimorfismo.

O TypeScript oferece então uma forma de corrigir ou contornar esses problemas no JS, adicionando funcionalidades que quando compiladas resultarão em código JavaScript novamente. Porém, o desenvolvedor lidará diretamente com uma sintaxe simplificada, mais clara e amplamente suportada por editores de código modernos.

## **1.2 Abas**

Organizado com o layout de abas o software foram desenvolvidos com 4 (quatro) divisões principais, sendo elas: Eventos, Cursos, Locais e Informações respectivamente. Dispostas de forma ergonômica onde a sessão mais usada está posicionada em um local que o clique fique mais confortável. Cada uma delas possui uma navegação individual a fim de receber os dados de forma mais organizada. Cada aba tem sua função única no software, para tornar a experiência de uso mais prática.

### **1.2.1 Eventos**

Com o intuito de informar o usuário de eventos, provas e procedimentos administrativos diversos, como exemplo, renovação do selo da carteirinha do período letivo, aditamento do FIES disponível, período de entrega de trabalhos e ED's, contexto geral ela é necessária para informações temporárias da instituição.

São exibidos em componentes em formato de Cards e possuem as seguintes informações: Título, descrição, data (de – até) e local. Recebe informações do provider “eventos” que permite a comunicação com a tabela “e\_data” do Banco de Dados e possui as integrações de Leitura, para usuários comuns (clientes) e leitura, edição, inclusão e exclusão de registros para usuários Administradores.

### **1.2.2 Cursos**

Com design remodelado, a aba Cursos possui uma navegação onde em sua tela principal são exibidos os dados fixos do aluno de acordo com sua configuração. Uma área de Consulta Rápida foi implementada para o aluno buscar outros cursos de seu interesse.

A consulta é feita por meio da escolha do Turno, o sistema exibe do Banco de dados por meio do provider “cursos” na tabela “c\_data” todos os cursos oferecidos da escolha, então ao escolher o curso a requisição é feita do caminho ‘c-data/PERÍODO ESCOLHIDO/CURSO ESCOLHIDO’ e então é necessária a seleção do semestre do mesmo.

Possui as integrações de Leitura, para usuários comuns (clientes), e Leitura, Edição, Inclusão e Exclusão de registros para usuários Administradores.

### 1.2.3 Locais

Centro direito. Contendo uma listagem de pontos de referência do campus, como exemplo: bibliotecas, auditórios e laboratórios em seus respectivos blocos e andares, facilitando o reconhecimento do local atribuído da pesquisa.

Com uma navegação única de “Locais/local” o cliente navega entre a tabela “l\_data” por meio do provider “locais”, onde é requisitado do Banco de dados uma listagem de todos os locais cadastrados e após o acesso a um ponto específico é exibido suas informações em componentes Cards contendo nome do local, descrição, serviços prestados, capacidade e horário de funcionamento. Possui as integrações de Leitura, para usuários comuns (clientes), e Leitura, Edição, Inclusão e Exclusão de registros para usuários Administradores.

### 1.2.4 Informações

Esta aba tem funções específicas para dúvidas referentes ao aplicativo e universidade com aplicações de contate-nos, tire suas dúvidas e criadores do aplicativo. Com a função de reconfiguração do perfil de usuário para resetar os valores atribuídos nas variáveis e tornar um novo curso como fixo.



## 2 WEB SERVICE

Realizando a comunicação entre os servidores da Google Firebase, a página administrativa e o cliente, onde são requisitados os dados necessários que são recebidos por JSON e injetados no sistema através de Providers.

Os Providers são responsáveis pela tradução dos dados recebidos e pelas funções solicitadas. No caso do usuário final a função solicitada pelo provider é somente de leitura, já as funções requisitadas pela página administrativa são de autenticação, criação de usuário, leitura, criação de registros, atualização e exclusão.

Ao todo foram desenvolvidos 5 (cinco) providers exclusivos para o projeto: eventos.provider.ts - responsável pela aba de eventos e pela comunicação com a tabela “e\_data”.

cursos.provider.ts - responsável pela aba de cursos e pela comunicação com a tabela “c\_data”.

locais.provider.ts - responsável pela aba de locais e pela comunicação com a tabela “l\_data”.

auth-service.provider.ts - responsável pela comunicação de segurança de usuários da página administrativa, realizando o cadastro, login e serviço de recuperação de senha.

Auth - responsável pela variável de conexão. Possuindo as chaves e id's únicos de identificação do servidor e Banco de dados.

Sendo somente o “auth-service.provider.ts” o único que não é implantado no aplicativo pois ele trata das funções de gerenciamento de usuários da página administrativa que não está disponível no cliente mobile.

Cada unidade instalada recebe uma chave de acesso única através dos providers que indica o banco de dados a ser acessado, o número de acesso, seu ID e sua permissão concedida previamente (se usuário – acesso de leitura, se root – acesso total) impedindo assim que softwares externos façam modificações ou obtenham as informações armazenadas, garantindo a integridade dos dados.

O sistema administrativo Uniinfo possui uma plataforma completa de controle dos dados e de usuários com funcionalidades exclusivas de recuperação de senha de administrador, que envia um e-mail com os dados de cadastramento de senha ao usuário caso este tenha a perdido, garantindo assim que seu acesso sempre

esteja seguro e garantido pelos servidores da internacionalmente conhecida Google LLC recurso de cadastro de novos usuários root.

Através de um usuário administrador previamente cadastrado é possível incluir novos gerentes, o que garante o compartilhamento de tarefas de implantação de dados, aumentando consideravelmente a eficiência e a qualidade das informações publicadas e consumidas pelos clientes.

Único com a tecnologia Real-Time Database que permite o recebimento instantâneo pelo aplicativo das informações publicadas, colocando o Uniinfo como o software mais ágil do campus, com um atraso máximo de apenas 2 (dois) segundos entre publicação e leitura.

## **2.1 Banco de Dados**

Para o sistema foi utilizado um Banco de Dados exclusivo graças a sua requisição de dados específicos como os de Eventos e Locais. O Firebase, oferecido pela Alphabet Inc. Organização que administra a empresa Google LLC, é um banco de dados online que disponibiliza soluções profissionais e especializadas em plataformas WEB e Mobile para gerenciamento e controle de dados. Aplicativos como The New York Times, Shazam, Duolingo, Alibaba.com, Lyft, Venmo, The Economist e Trivago tem como sua base de dados o Firebase. Utilizando a autenticação de usuários por e-mail, o sistema garante o acesso controlado de administradores e Evita que os usuários criem várias contas usando o mesmo endereço de e-mail com provedores diferentes de autenticação.

O recurso de cote de inscrição limitamos o cadastro de novos roots pelo mesmo endereço IP para apenas dois cadastros/hora, evitando assim um ataque em grande escala de login's falsos. Através do sistema Firebase Analytics Events, é possível ter o log de acesso e modificações do banco. Pelo Firebase Analytics Audiences temos as informações de uso e performance.

Com o Firebase Develop Database Rules definimos o tipo de autorização de cada usuário (se root – se usuário). E com o Firebase Develop Database Backup é feito diariamente um Backup Incremental e semanalmente um Backup Full previamente programado, armazenado na nuvem da Google e garantindo a maior integridade dos dados.



O método de comunicação com o servidor é feito por providers internos do aplicativo, onde a requisição por meios das funções específicas, por exemplo, o `Get()` é percorrida do aplicativo em HTML é enviado para o TypeScript onde é definido o tipo de requisição que será feita do provider, que então é solicitado através de um ID e uma chave única que direciona ao banco de dados do Firebase. Após essa requisição o aplicativo armazena a informação em cache para demais consultas mesmo off-line.



### 3 HEURISTICAS

Heurística é uma técnica onde você torna o produto software mais usável para seu cliente, fazendo com que o sistema tenha um sentido cognitivo para o usuário. Heurística por definição é um "atalho mental" usado no pensamento humano para se chegar a resultados rápidos, mesmo que estes sejam incertos ou incompletos. Um parâmetro desenvolvido por Jakob Nielsen visa deixar a interface do sistema mais usável, o método é conhecido como 10 heurísticas e traz dez parâmetros que podem ser aplicados no software e melhorar a experiência do usuário.

#### 3.1 Visibilidade do estado do sistema

É responsabilidade do sistema informar o que está acontecendo em tempo real ao usuário, informando o status de onde ele está no sistema, facilitando entendimento do usuário de onde está de onde veio e para onde pode ir.

Imagem 2 – Consulta Rápida

← **Consulta Rápida**  
Cursos > Consulta Rápida

Consulta de localização de salas para outros cursos.

Busca por filtro  
Turno | Curso | Semestre

\* Turno: ▾  
\* Curso: ▾  
\* Semestre: ▾

Campos com (\*) são de preenchimento obrigatório.

VER SALA 🔍  
Corrigir

Eventos Cursos Locais Mais

Fonte – Uniinfo 2018

### 3.2 Correspondência entre o sistema e o mundo real

É o relacionamento do software com o mundo real a compatibilidade com essas externalidades: ícones como uma casa, ou um disquete refletem ideias do cotidiano, onde a casa seria o botão home e o disquete o botão de salvar.

Imagem 3 – Ícone



Fonte – Uniinfo 2018

### 3.3 Liberdade de controle fácil ao usuário

A preocupação é de passar para o usuário a liberdade de ele fazer o que quiser dentro do sistema com exceção das regras que vão contra o negócio ou interferem em outra funcionalidade, a possibilidade de reverter ações diminui a insegurança do usuário ao utilizar o aplicativo, a liberdade de controle seria dar o "ctrl+Z" ao usuário.

Imagem 4 – Botão Corrigir

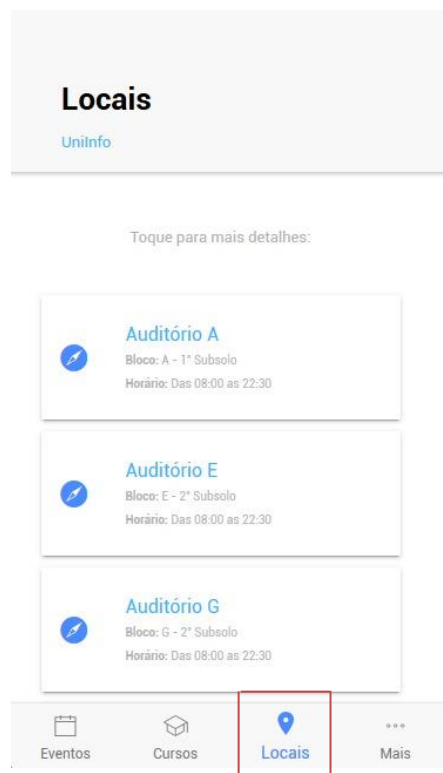
A interface de consulta rápida de salas de aula do aplicativo Uniinfo 2018. No topo, há uma barra de navegação com um ícone de seta para trás e o título "Consulta Rápida", com o caminho "Cursos > Consulta Rápida" abaixo dele. Abaixo, uma sub-título indica "Consulta de localização de salas para outros cursos.". O formulário principal contém uma seção de busca com o texto "Busca por filtro" e "Turno | Curso | Semestre". Abaixo disso, há três campos obrigatórios com marcadores de asterisco: "Turno:", "Curso:" e "Semestre:", cada um com uma seta para baixo indicando uma lista suspensa. Uma mensagem de aviso abaixo dos campos diz: "Campos com (\*) são de preenchimento obrigatório.". Abaixo do formulário, há um botão azul "VER SALA" com um ícone de lupa e um botão "Corrigir" destacado por um retângulo vermelho. Na base da tela, há uma barra de navegação com quatro ícones: "Eventos" (calendário), "Cursos" (graduação, em azul), "Locais" (localização) e "Mais" (três pontos).

Fonte – Uniinfo 2018

### 3.4 Consistência e padrões

A consistência e padronização estão nos ícones padronizados com desenhos de fácil reconhecimento ao usuário, como a lupa o ícone de um livro na frente da barra do curso a cor padrão do aplicativo (azul e branco).

Imagem 5 - Simbolo



Fonte – Uniinfo 2018

### 3.5 Prevenções de erros

É melhor prevenir o erro do que remediá-lo, ou seja fornecer ao usuário um método que possa tira-lo de situações irreversíveis, como uma caixa de texto perguntando se realmente deseja executar tal ação.

Imagem 6 – Botão de Consulta Rápida

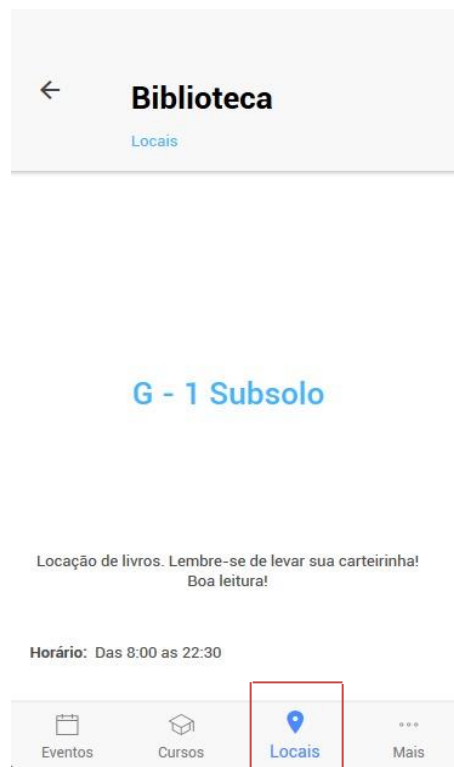


Fonte – Uniinfo 2018

### 3.6 Reconhecimento em vez de memorização

O uso de ícones como “casa”, botão de voltar entre outros, já são itens de reconhecimento onde o usuário já entende a funcionalidade de tal ação no sistema por já ter visto sua funcionalidade em outros sistemas, como o local como os cards de seleção, onde você clica e rola a tela até encontra o item buscado.

Imagem 7 – Ícone Aba Locais



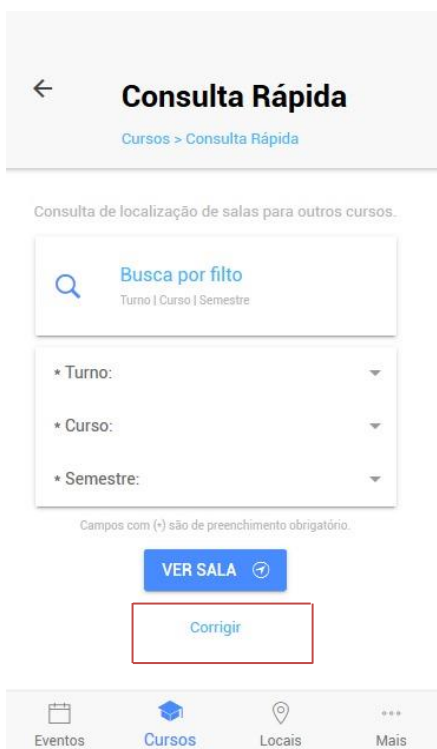
Fonte – Uniinfo 2018

### 3.7 Flexibilidade e eficiência de uso

A flexibilidade de uso está ligada a facilitar o uso do sistema para o usuário, é importante deixar uma boa experiência ao usuário desde o mais leigo até o mais avançado.




Imagem 8 – Corrigir Consulta Rápida



← **Consulta Rápida**  
[Cursos > Consulta Rápida](#)

Consulta de localização de salas para outros cursos.

 **Busca por filtro**  
Turno | Curso | Semestre

\* Turno: ▼  
\* Curso: ▼  
\* Semestre: ▼

Campos com (\*) são de preenchimento obrigatório.

**VER SALA** ↺

**Corrigir**

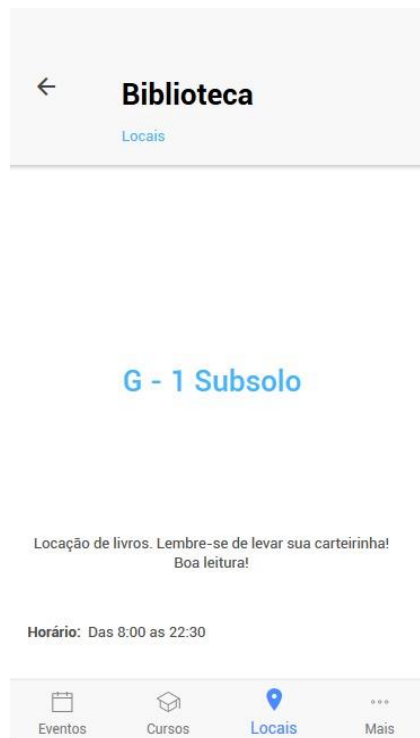
Eventos **Cursos** Locais Mais

Fonte – Uniinfo 2018

### 3.8 Estética e design minimalista

Toda informação extra que você deixar para o usuário pode na verdade adicionar mais uma dúvida, um conteúdo claro e um layout limpo tem mais a somar do que uma tela toda detalhada em muitos casos, informações secundárias são deixadas de lado como outras abas por exemplo, isso poderia gerar informações redundantes ou desnecessárias.

Imagem 9 – Aba locais

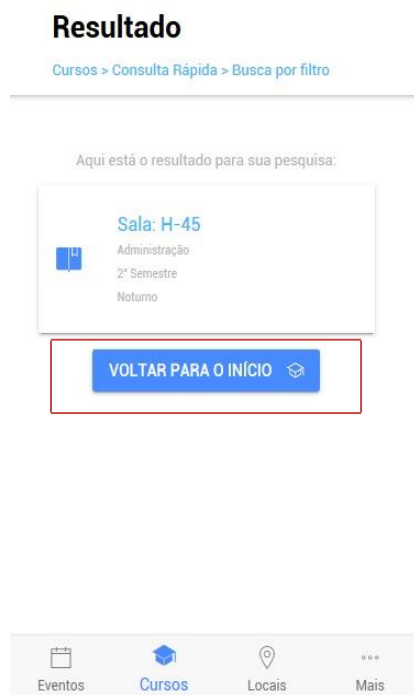


Fonte – Uniinfo 2018

### 3.9 Ajuda ao usuário a recuperar-se de erros

É normal cometer erros principalmente ao iniciar a usar algo novo, então quando o usuário fizer algo errado o sistema deve estar preparado para tratar o erro e até mesmo corrigi-lo se for necessário.

## Imagem 10 – Voltar ao início



Fonte – Uniinfo 2018

### 3.10 Ajuda e documentação

Ao abrir o aplicativo na primeira vez, o usuário tem telas que aparecem apenas na sua primeira utilização, que explicam quais são as funcionalidades de cada aba do aplicativo documentando suas funções.



## 4 UML

UML é uma linguagem que define uma série de artefatos que nos ajuda na tarefa de modelar e documentar os sistemas orientados a objetos que desenvolvemos, é a parte anterior ao projeto, muitas vezes é até mesmo subestimada por ser algo “simples”, mas na verdade é a idealização do projeto em modelos relevantes a construção do software.

Linguagem de modelagem unificada (UML) é uma linguagem visual para documentação de projeto e padrões de software, podendo ser aplicada em várias áreas diferentes. É uma relação de documentação e expressões, comportamentos e ideias numa notação que é fácil de entender e eficiente para se escrever. UML é visual, quase tudo nela tem uma representação gráfica facilitando o entendimento de todos os envolvidos. Em geral, um modelo UML, é feito de um ou mais diagramas.

Um diagrama representa graficamente coisas e as relações entre essas coisas. Essas coisas podem ser representações de objetos do mundo real, construções de software ou uma descrição de comportamento de algum outro objeto. Aprender a aplicar a UML pressupõe o entendimento de três elementos, Itens, relacionamentos e diagramas.

### 4.1 Caso de Uso

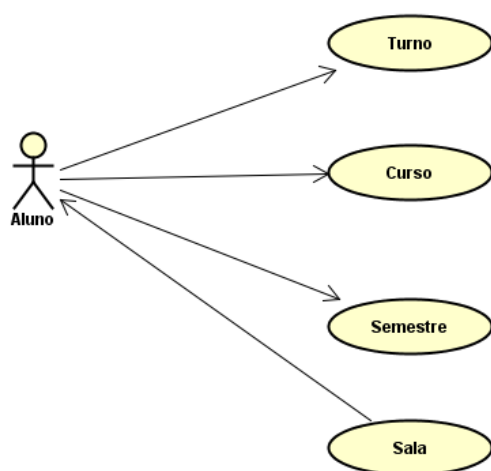
Esse diagrama documenta o que o sistema faz do ponto de vista do usuário. Em outras palavras, ele descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema de uma forma que qualquer pessoa possa entender o que deverá acontecer no sistema. Nesse diagrama os detalhes são superficiais, ou seja, não define em detalhes técnicos como o sistema faz. Um caso de uso tem basicamente quatro partes, ator, cenário, caso de uso e comunicação.

Nome do caso:	Localizar sala.
Atores	Aluno.
O que os atores esperam:	Um sistema eficiente e direto, sem erros.
Descrição do processo normal:	<p>Informar sala.</p> <p>Informar período.</p> <p>Informar semestra.</p> <p>Receber dados sobre a sala.</p>
Quando não localiza-se a sala:	<p>Consultar informações na secretaria.</p> <p>Enviar feedback.</p> <p>Aguardar atualização.</p>
Descrição dos possíveis erros:	<p>Erro ao receber dados:</p> <p>Informações imprecisas.</p> <p>Falta de dados.</p> <p>Anomalias geradas por Informações incorretas.</p> <p>Erro de conexão.</p>
Caso alternativo:	<p>Envio de informações via email.</p> <p>Conectar-se a internet.</p> <p>Enviar feedback.</p> <p>Aguardar nova atualização.</p> <p>Consultar a secretaria do campus</p>
Resultado final do caso:	Receber dados sobre sua sala.
Sala não encontrada:	<p>Envio de formulário.</p> <p>Envio de feedback.</p>

#### 4.1.1 Diagrama de Caso de Uso

O diagrama do caso de uso é a modelagem do caso de uso e um diagrama com desenhos e representações que exemplificão com praticidade quem são os atores e quais suas interações com o sistema uma vez definido o cenário

Imagem 11 – Diagrama de caso de Uso



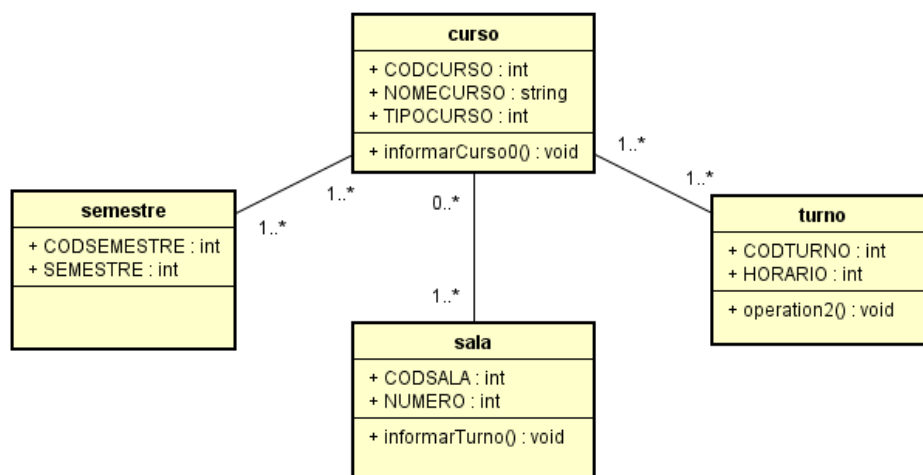
Fonte – Uniinfo 2018

#### 4.1.2 Diagrama de Classe

O diagrama de classes é utilizado na construção do modelo de classes desde o nível de análise até o nível de especificação. Cada classe tem por definição, Nome, Atributos e Métodos.

Uma classe é representada por uma “caixa” com, no máximo, três compartimentos exibidos. No primeiro compartimento (de cima para baixo) é exibido o nome da classe. Por convenção, esse nome é apresentado no singular e com as palavras componentes começando por maiúsculas. No segundo compartimento, são declarados os atributos, que correspondem às informações que um objeto armazena. Finalmente, no terceiro compartimento, são declaradas as operações, que correspondem às ações que um objeto sabe realizar.

Imagem 12 – Diagrama de Classe

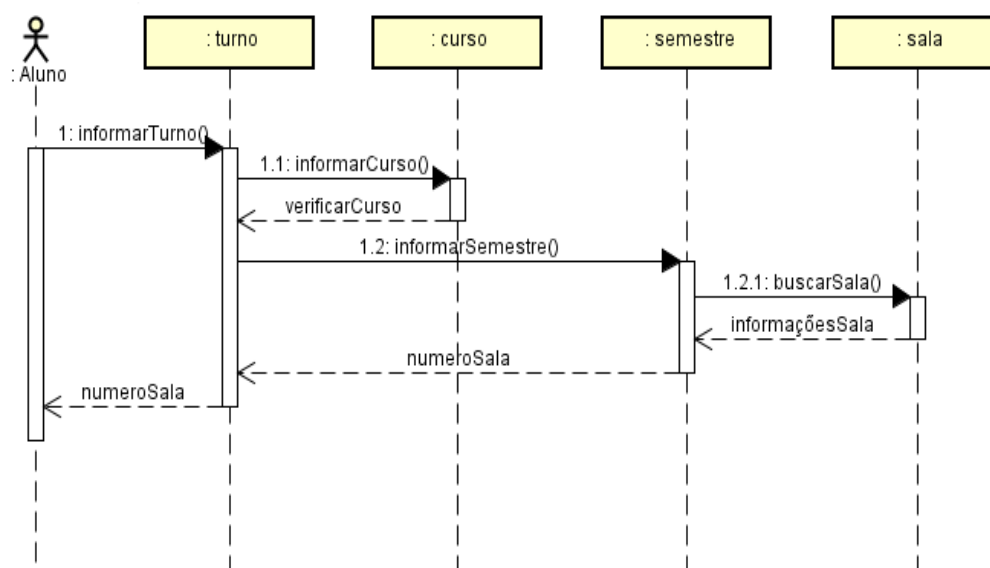


Fonte – Uniinfo 2018.

#### 4.1.3 Diagrama de sequência

O diagrama de sequência visa fornecer visualmente a cadeia de processos de um caso de uso, mostrando passo a passo as interações dos objetos entre si em determinado cenário, desde o início até o fim do processo.

Imagem 13 – Diagrama de Sequência



Fonte – Uniinfo 2018.



## **5 ANÁLISE DE RISCOS**

A análise de riscos consiste em tentar prever os eventos possíveis, avaliando toda e qualquer probabilidade de que o projeto desenvolvido venha a apresentar erros, a partir desta análise é efetuado o levantamento das ameaças e desenvolve medidas preventivas, ou seja, se houver erro no software seriam essas as medidas aplicáveis para sua resolução, ou até mesmo uma maneira de diminuir seu impacto.

Para fazer a análise de risco do software é necessário levantar os possíveis erros que podem ocorrer, ou seja, entender o que poderá acontecer no sistema para que possam ser desenvolvidos métodos de solucionamento dos problemas. Em seguida formula-se a quantificação do risco, a probabilidade de ocorrência e seu impacto no funcionamento do sistema, determinando o quão considerável para o funcionamento do software é o erro.

No final da análise de risco desenvolvem-se procedimentos que deverão ser realizados para a resolução do problema encontrado, dessa maneira evitando o máximo de erros que puderem ser previstos pelos desenvolvedores antes do término do projeto.

### **5.1 Análise de ameaças**

É o estudo das causas das ameaças e de suas consequências sobre o projeto. Avaliam-se as chances de reincidência e os efeitos produzidos, em caso de novas ocorrências.

Neste processo são levantadas todas as possíveis ameaças, para descobrir os erros que podem vir a afetar o software são feitas diversas reuniões para a aquisição de um maior inventário de prováveis ameaças.

### **5.2 Avaliação dos riscos**

É a segregação dos riscos conforme o nível do impacto produzido, podendo ser alto, médio ou baixo. Há diversos tipos de classificação em níveis: baixo/pequeno, médio/moderado e alto/grande.

Outro caminho para realizar a análise é segmentar o risco em probabilidade e impacto. O impacto é quanto o sistema será afetada ao ser exposto a determinado

problema e probabilidade é a quantificação provável de tal acontecimento. Ao combinar esses métodos, pode-se produzir uma matriz de avaliação de probabilidade e de impacto de riscos.

### **5.3 Tratamento dos riscos**

Tratar riscos é modificar a situação atual para que o cenário empresarial não fique mais propício ao problema identificado.

Nesse momento são desenvolvidos métodos com o objetivo de minimizar as chances de que um determinado erro aconteça e prejudique o projeto, maximizando a qualidade do produto final.

Diversas soluções devem ser formadas para o tratamento do risco, a que se adequar melhor ao plano do projeto sem depreciar o projeto final é que deverá ser aplicada.

## **6 PADRÕES DE QUALIDADE - ISO/IEC 9126 (NBR 13596)**

Uma norma que tem como objetivo avaliar a qualidade, bem como as suas características, subcaracterísticas e atributos. Este é o objetivo da ISO 9126, mais especificamente a NBR 13596, que padroniza a avaliação da qualidade do software. Um produto final com qualidade é uma necessidade, uma vez que desde o levantamento de requisitos até a fase de produção, os fatores que atendem ao que o cliente precisa devem estar em evidência, bem como o feedback para melhoria contínua dos processos da organização.

### **6.1 Introdução ISO/IEC 9126**

A ISO/IEC 9126 é uma norma ISO para qualidade de produto de software, ela define um conjunto de parâmetros com o objetivo de padronizar a avaliação da qualidade de software. Ela se enquadra no modelo de qualidade das normas da família 9000. A norma brasileira correspondente é a NBR 13596, que foi substituída pela NBR ISO/IEC 9126-1.

### **6.2 Modelo de qualidade de software**

A qualidade de um sistema de software pode ser entendida de diversas formas e utilizando diferentes abordagens.

A norma ISO/IEC 9126, ou conjunto de normas que tratam deste assunto no âmbito da ISO, estabelece um modelo de qualidade com os seguintes componentes:

#### **6.2.1 Processo de desenvolvimento**

Cuja qualidade afeta a qualidade do produto de software gerado e é influenciado pela natureza do produto desenvolvido.

#### **6.2.2 Produto**

Compreendendo os atributos de qualidade do produto (sistema) de software. Estes atributos de qualidade podem ser divididos entre atributos internos e externos.

Estes se diferenciam pela forma como são aferidos (interna ou externamente ao produto de software) e em conjunto compõem a qualidade do produto de software em si.

### **6.3 Qualidade em uso**

Que consiste na aferição da qualidade do software em cada contexto específico de usuário. Esta é, também, a qualidade percebida pelo usuário.

A norma 9126 foca na qualidade do produto de software, propondo Atributos de Qualidade, distribuídos em seis características principais, com cada uma delas divididas em subcaracterísticas: Funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.

#### **6.3.1 Funcionalidade (satisfaz as necessidades?)**

A capacidade de um software prover funcionalidades que satisfaçam o usuário em suas necessidades declaradas e implícitas, dentro de um determinado contexto de uso. Suas subcaracterísticas são:

##### **6.3.1.1 Adequação**

Que mede o quanto o conjunto de funcionalidades é adequado às necessidades do usuário, ou seja, propõe-se a fazer o que é apropriado? Importante ressaltar o trabalho do analista de sistemas nesta categoria, pois depende muito da sua competência entender e especificar o que o usuário precisa em um software.

##### **6.3.1.2 Acurácia**

Representa a capacidade do software de fornecer resultados precisos ou com a precisão dentro do que foi acordado/solicitado, ou seja, Faz o que foi proposto de forma correta? Ligado a Adequação, representa, no sumo, as regras de negócio da aplicação.

##### **6.3.1.3 Interoperabilidade**

Que trata da maneira como o software interage com outro(s) sistema(s) especificado(s), ou seja, Interage com os sistemas já existentes, ou no caso se for um subsistema, quão bem interage com o supersistema.

#### 6.3.1.4 Segurança

Mede a capacidade do sistema de proteger as informações do usuário e fornecê-las apenas (e sempre) às pessoas autorizadas, ou seja, Evita acesso não autorizado aos dados e garante o acesso autorizado?

#### 6.3.2 Confiabilidade (é imune a falhas?)

O produto se mantém no nível de desempenho nas condições estabelecidas. Suas subcaracterísticas são:

##### 6.3.2.1 Maturidade

Entendida como sendo a capacidade do software em evitar falhas decorrentes de defeitos no software (Com que frequência apresenta falhas?). Sabemos que nenhum software está livre de bugs, mas podemos criar mecanismos para gerenciar erros em tempo de execução, a fim de minimizar os problemas causados por bugs não previstos (podendo incluir geração de logs, tratamento de exceções, backups e etc).

##### 6.3.2.2 Tolerância a falhas

Representando a capacidade do software em manter o funcionamento adequado mesmo quando ocorrem defeitos nele ou nas suas interfaces externas, ou seja, ocorrendo falhas, qual a reação do sistema? Garantir que o resto do sistema não pare de funcionar por conta de somente um pedaço do sistema que gerou problemas.

### 6.3.2.3 Recuperabilidade

Que foca na capacidade de um software se recuperar após uma falha, restabelecendo seus níveis de desempenho e recuperando os seus dados, ou seja, É capaz de recuperar dados em caso de falha? Principalmente ligado a banco de dados. Backups!

### 6.3.3 Usabilidade (é fácil de usar?)

A capacidade do produto de software ser compreendido, seu funcionamento aprendido, ser operado e ser atraente ao usuário.

Note que este conceito é bastante abrangente e se aplica mesmo a programas que não possuem uma interface para o usuário final. Por exemplo, um programa batch executado por uma ferramenta de programação de processos também pode ser avaliado quanto a sua usabilidade, no que diz respeito a ser facilmente compreendido, aprendido, etc. Além disto, a operação de um sistema é uma interface Humano-Computador (ver IHC) sujeita às avaliações de usabilidade.

Suas subcaracterísticas são:

#### 6.3.3.1 Inteligibilidade

Que representa a facilidade com que o usuário pode compreender as suas funcionalidades e avaliar se o mesmo pode ser usado para satisfazer as suas necessidades específicas, ou seja, É fácil entender o conceito e a aplicação?

#### 6.3.3.2 Apreensibilidade

Identifica a facilidade de aprendizado do sistema para os seus potenciais usuários, ou seja, É fácil aprender a usar?

#### 6.3.3.3 Operacionalidade

É como o produto facilita a sua operação por parte do usuário, incluindo a maneira como ele tolera erros de operação, ou seja, É fácil de operar e controlar?

Atratividade envolve características que possam atrair um potencial usuário para o sistema, o que pode incluir desde a adequação das informações prestadas para o usuário até os requintes visuais utilizados na sua interface gráfica.

#### 6.3.4 Eficiência (é rápido e enxuto?)

O tempo de execução e os recursos envolvidos são compatíveis com o nível de desempenho do software. Suas subcaracterísticas são:

##### 6.3.4.1 Comportamento em relação ao tempo

Que avalia se os tempos de resposta (ou de processamento) estão dentro das especificações, ou seja, qual o tempo de resposta, a velocidade da execução?

##### 6.3.4.2 Utilização de recursos

Que mede tanto os recursos consumidos quanto a capacidade do sistema em utilizar recursos disponíveis. Quantos recursos são usados, durante quanto tempo?

#### 6.3.5 Manutenibilidade (é fácil de modificar?)

A capacidade (ou facilidade) do produto de software ser modificado, incluindo tanto as melhorias ou extensões de funcionalidade quanto as correções de defeitos, falhas ou erros. Suas subcaracterísticas são:

##### 6.3.5.1 Modificabilidade

Caracteriza a facilidade com que o comportamento do software pode ser modificado, o quão fácil ele é de modificar. Sistemas com baixo nível de abstração são muito mais difíceis de modificar, tanto para adicionar novas funcionalidades quanto para alterar as já existentes sem influenciar o comportamento do resto da aplicação. Aplicações “presas”.

##### 6.3.5.2 Estabilidade

Avalia a capacidade do software de evitar efeitos colaterais decorrentes de modificações introduzidas, ou seja, o que garante que o software continuará funcionando corretamente com modificações futuras. O banco aguentará? Testes foram criados para garantir que o que já está pronto não será perdido (testes unitários)?

#### 6.3.5.3 Testabilidade

Representa a capacidade de se testar o sistema modificado, tanto quanto as novas funcionalidades quanto as funcionalidades não afetadas diretamente pela modificação (testes unitários, de novo).

#### 6.3.6 Portabilidade (é fácil de migrar para outro ambiente?)

Portabilidade é a capacidade de o sistema ser transferido de um ambiente para outro. Como "ambiente", devemos considerar todos os fatores de adaptação, tais como diferentes condições de infraestrutura (sistemas operacionais, versões de bancos de dados, etc.), diferentes tipos e recursos de hardware (tal como aproveitar um número maior de processadores ou memória). Além destes, fatores como idioma ou a facilidade para se criar ambientes de testes devem ser considerados como características de portabilidade. Suas subcaracterísticas são:

##### 6.3.6.1 Adaptabilidade

Representando a capacidade do software de se adaptar a diferentes ambientes sem a necessidade de ações adicionais (configurações). Exemplos: Ambientes (Windows, Linux, Mac; Bancos Oracle, Sql, Db2).

##### 6.3.6.2 Capacidade para ser instalado

Identifica a facilidade com que pode se instalar o sistema em um novo ambiente;



#### 6.3.6.3 Coexistência

Mede o quão facilmente um software convive com outros instalados no mesmo ambiente (Irá gerar conflitos?).

#### 6.3.6.4 Capacidade para substituir

Representa a capacidade que o sistema tem de substituir outro sistema especificado, em um contexto de uso e ambiente específicos. Este atributo interage tanto com adaptabilidade de quanto com a capacidade para ser instalado. Geralmente, no desenvolvimento de um software substituto para um sistema ultrapassado, chega um momento na homologação do novo sistema onde os dois sistemas, tanto o novo quanto o antigo, ficarão em produção simultaneamente, para garantir que falhas não previstas no novo sistema aconteçam e ocorra a perda de dados significativos. O sistema antigo será desativado gradualmente, conforme o sistema novo for “otimizado”.

#### 6.3.6.5 Analisabilidade

Identifica a facilidade em se diagnosticar eventuais problemas e identificar as causas das deficiências ou falhas, ou seja, a facilidade em se encontrar falhas no sistema, quando elas ocorrerem (lembrando que nenhum sistema é imune a falhas).

### 6.4 Conclusão

Os Desenvolvedores de Software fornecem os Softwares para os usuários usarem, devido a isso, surge a grande necessidade de criar formas para evitar o grande prejuízo decorrente de mal funcionamento de softwares e também de problemas (não intencionais) gerados pelos usuários. Com isso, vimos que só é possível garantir a qualidade de um software, caso o software em questão atenda as quatro características de Qualidade de Software, que são: Usabilidade, Confiabilidade, Funcionalidade e Manutenibilidade. Tendo em vista que estes são os requisitos exigidos pelos clientes e atendidos pelos fornecedores de Softwares. Seguindo estes, o resultado não poderá ser ruim. Teremos pontos positivos tanto

para quem produz o software, quanto para quem utiliza o software, evitando assim tais gastos elevadíssimos gerados por má qualidade (Sem que algum ou todos os requisitos de Qualidade sejam atendidos), o Uniinfo segue as normas de qualidade ISO 9126, buscando atender os requisitos da mesma e entregar um produto de alta qualidade para o usuário final.

## **7 CONCLUSÃO**

Conclui-se que o projeto Uniinfo será de grande estima como um software de ajuda/consulta, que não atende apenas a classe estudantil. Atualizações deverão ser feitas corriqueiramente para que a usabilidade e a informação contida no projeto estejam corretas. Os produtos derivados do projeto têm relações com os ensinamentos aplicados pela UNIP no curso de ADS, ao todo desde a concepção até o fim do projeto dezoito meses foram utilizados em seu desenvolvimento, e o aplicativo encontra-se em sua terceira versão.



## REFERÊNCIAS BIBLIOGRÁFICAS

Astah – Astah Download – Astah, Internet, 2018. Disponível em: <<http://astah.net/download>>. Acessado 03/05/18.

(Aula 1) Prática - UML Diagrama de classes – (Atributos, Operações e Herança). Soares, Sthefane. Youtube. 02/10/2014 12min2s. Internet, disponível em: <<https://www.youtube.com/watch?v=9Wdjuds5vPQ>>. Acessado 02/05/18.

Aula1 - Modelagem UML - Diagrama de Caso de Uso, Classe e Sequência. Brenner, Gabriel. Youtube. 21/06/2015 41min58s. Disponível em: <<https://www.youtube.com/watch?v=pCCqjJDLr4Y>> Acessado em 20/04/18.

Google – Ajuda do Firebase. Google, Internet, 2018. Disponível em: <<https://support.google.com/firebase/answer/6400716>>. Acessado em 25/04/18.

JOSÉ, Edson - DEVMEDIA – Introdução ao TypeScript, 2016. Internet, disponível em: <<https://www.devmedia.com.br/introducao-ao-typescript/36729>>. Acessado 27/03/18.

Microsoft – Diagramas de sequência UML: referência. Microsoft, internet 2018. Disponível em: <<https://msdn.microsoft.com/pt-br/library/dd409377.aspx>>. Acessado 13/05/18.

Ribeiro, Leandro - O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML. DEVMEDIA, internet 2012. Disponível em: <<https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>>. Acessado 13/05/18.

Strong Security - Análise de riscos em TI: o que é, como fazer e mais!. Strong Security, internet 2018. Disponível em: <<https://www.strongsecurity.com.br/analise-de-riscos-em-ti-o-que-e-como-fazer-e-mais/>>. Acessado 10/05/18.

Ruggieri, Ruggero – Análise sobre a ISO 9126 – NBR 13596. TI Especialistas, Internet 2016. Disponível em: <<https://www.tiespecialistas.com.br/analise-sobre-iso-9126-nbr-13596/>>acessado em 18/05/2018.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR ISO-9126: Engenharia de software*. Rio de Janeiro, 2003.

Gaea – O guia definitivo de gestão de riscos para TI. Gaea, Internet 2017. Disponível em: <<https://gaea.com.br/o-guia-definitivo-de-gestao-de-riscos-para-ti/>>acessado em 10/05/2018.



## APÊNDICE





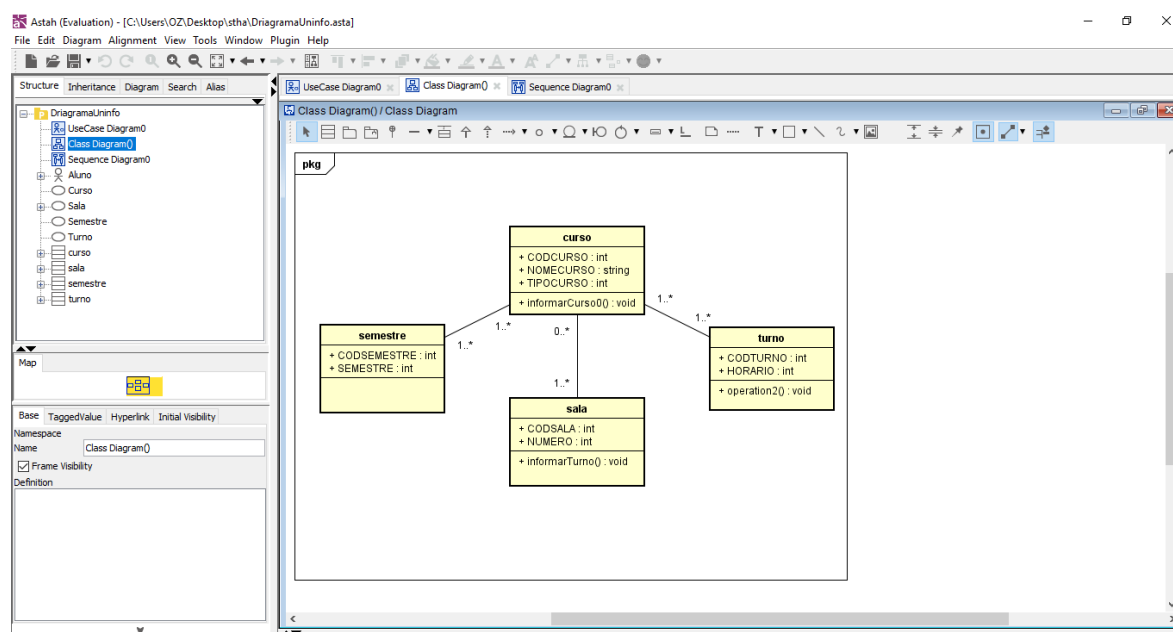
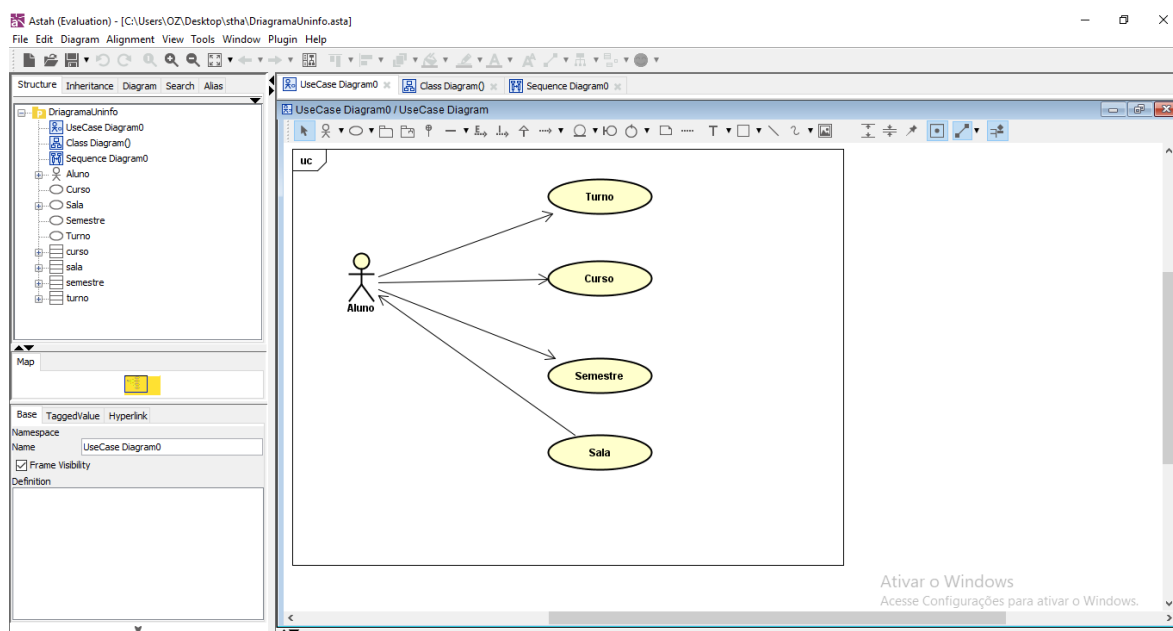
## APÊNDICE A - ANÁLISE DE RISCOS

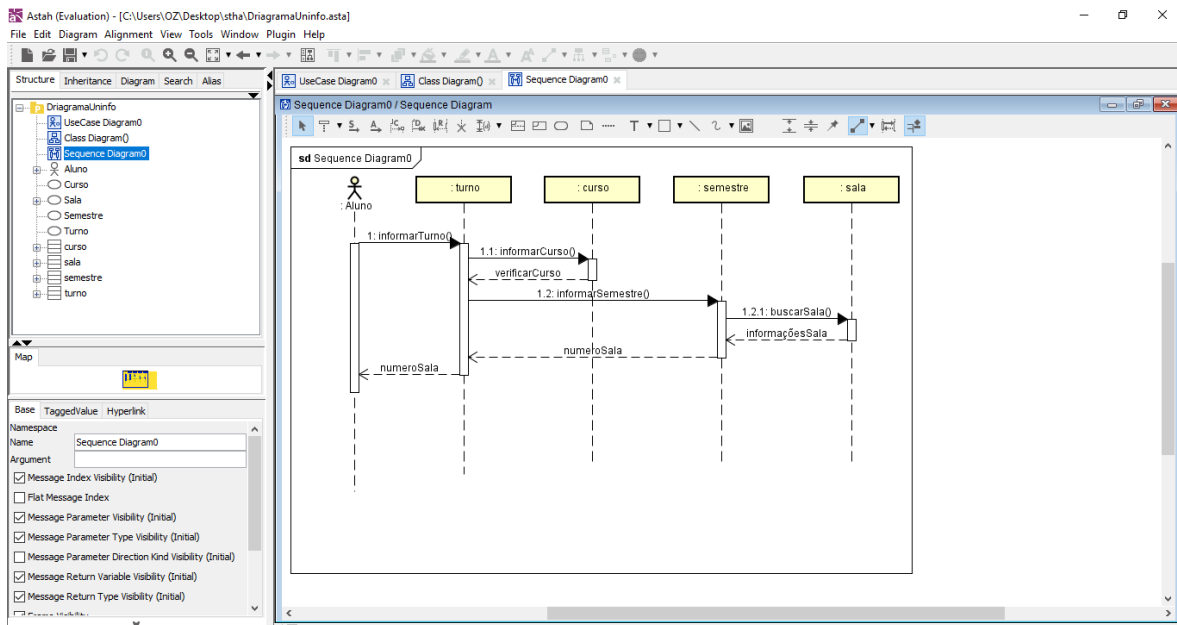
Análise de Riscos				
Ameaças	Risco	Probabilidade	Impacto	Solução
Lista de Cursos vazio	Médio	30%	Médio	Feche a janela do Aplicativo, verifique a conexão com a internet. Caso o erro persista, reinicie o aplicativo
Impossível avançar de Tela	Baixo	30%	Médio	Feche a janela de Seleção, abra novamente, e selecione um novo valor. Caso o erro persista, reinicie o aplicativo.
Não Encontrei um evento Determinado	Baixo	20%	Baixo	Todos os eventos são cadastrados pela equipe da secretaria, sendo assim entre em contato com seu coordenador
Dados do Evento Incorretos	Baixo	20%	Baixo	Todos os eventos são cadastrados pela equipe da secretaria, sendo assim entre em contato com seu coordenador
As informações do meu Curso Não aparecem	Baixo	30%	Médio	As informações do seu curso são gravadas de acordo com o passo-a-passo inicial do aplicativo, ou através da seção alterar meu perfil, caso após a configuração continue em branco, reinicie o aplicativo e refaça a configuração
Botão Corrigir não funciona	Baixo	10%	Baixo	Este erro se deve a um erro interno de reinitialização dos dados, por favor reinicie o aplicativo e tente novamente
Botão de ver sala não funciona corretamente ou me traz os dados incorretos	Baixo	10%	Médio	Isso se deve a um erro interno de gravação de dados, por favor reinicie o aplicativo e tente novamente
Alterar minhas Informações não funciona	Baixo	15%	Baixo	Este erro é causado por alguma falha na sincronização interna do aplicativo. Por favor reinicie o aplicativo e tente novamente.
Travamento do Aplicativo	Médio	20%	Médio	Por favor feche totalmente o aplicativo e reinicie ele.
Preencher dados sem ter selecionado Valores	Médio	20%	Médio	Caso na sua configuração de Perfil ele permita que voce avance as telas sem ter de selecionar, as informações, reinicie o aplicativo

Ativar o Windows



## APÊNDICE B – DIAGRAMAS UML





## APÊNDICE C – TELAS DO UNIINFO



### Eventos

Agora você tem um portal direto entre a secretaria e você! Saiba tudo que acontece pelo Campus. Datas, eventos, prazos e muitas outras informações importantes filtradas exclusivamente para seu curso.





## Cursos

Chega de pegar filas ou ficar perdido no começo do semestre procurando em um monte de papeis sua sala! Saia da sua casa já sabendo exatamente pra onde vai!

Calma, se tiver alguma daquelas mudanças chatas durante o semestre é só abrir seu App e em menos de 2 segundos a informação vai estar atualizada na sua tela!

A UnilInfo também é consciente, evite papeis!





## Locais

"Próxima aula será na nossa quadra de esportes." Sim, temos uma! Piscinas, três auditórios muito grandes e diversas salas! Perdido? Também te ajudamos nisso, basta acessar nossa seção exclusiva de locais para ver tudo que seu campus proporciona à você.





U n i  n f o

CONFIGURAR MEU PERFIL 







## Meu turno

Para personalizarmos seu ambiente precisamos que  
você selecione o **turno** do seu curso:

MANHÃ

NOITE

[Ajuda](#)



## Meu curso

Agora vamos precisar do seu **curso**:

Estudo:



PRÓXIMO >

[Ajuda](#)

\*Caso seu curso não esteja na lista, entre em contato com a secretaria de seu Campus.



## Meu semestre

Por fim... Seu **semestre**:

Estou no:



PRÓXIMO >

[Ajuda](#)

\*Caso seu semestre não esteja na lista, entre em contato com a secretaria de seu Campus.



## Boas vindas!

Agora confirme as informações abaixo para poder usar  
nossa interface personalizada para você!



Matiturno



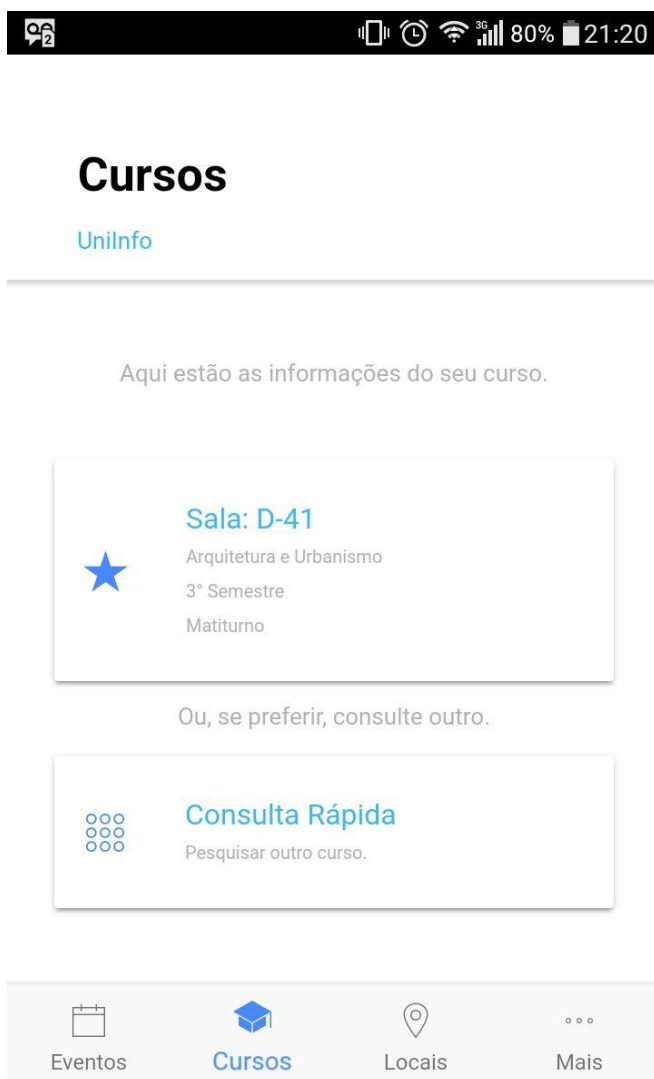
Arquitetura e Urbanismo

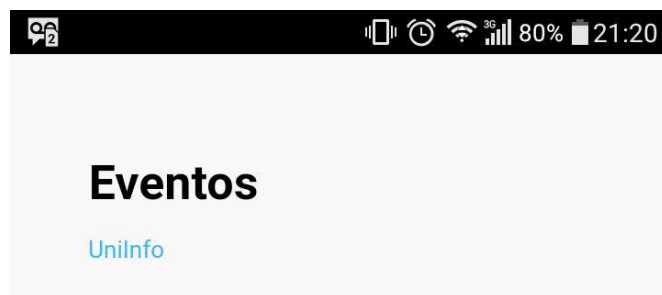


3° Semestre

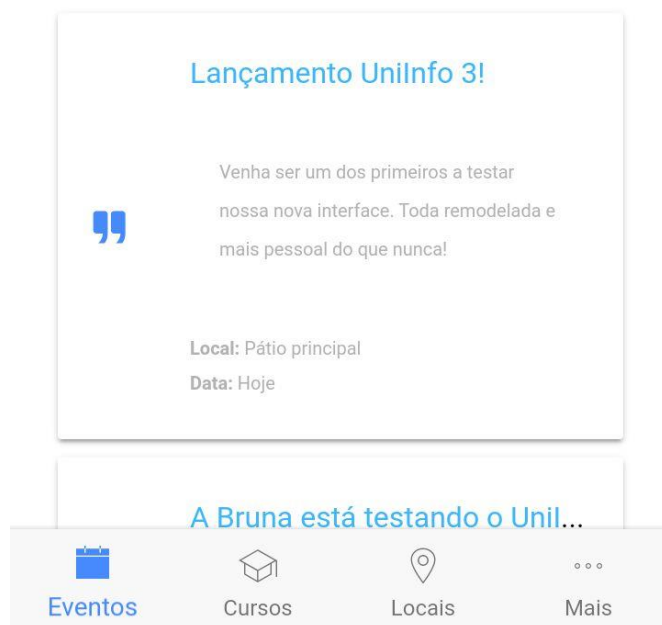
TUDO CERTO! ➔

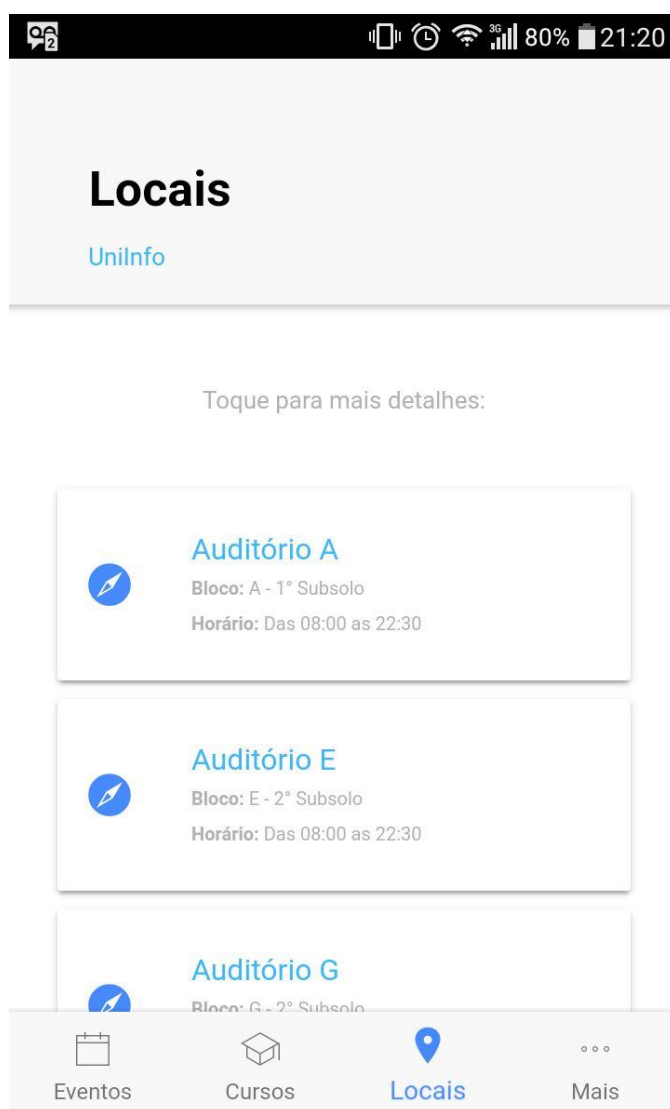
[Voltar e corrigir.](#)





Fique por dentro do que está acontecendo pelo campus:

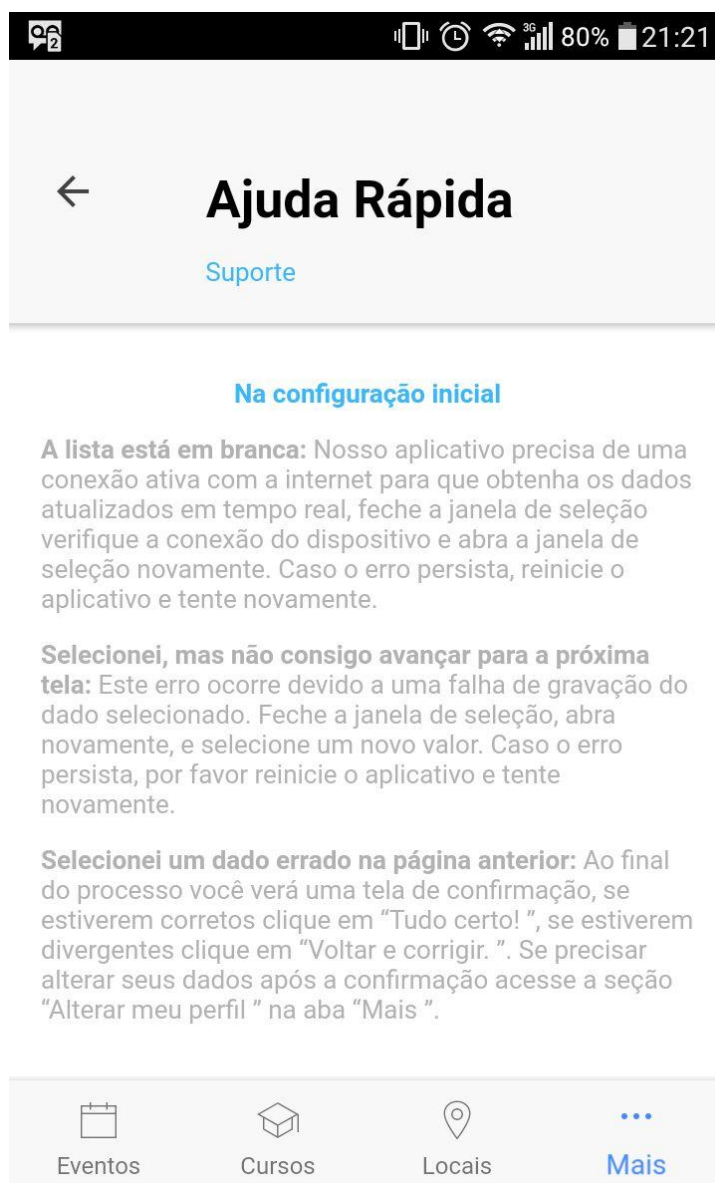












## APÊNDICE D - CÓDIGO FONTE APLICAÇÃO WEB

```
app.components.ts

import { Component } from '@angular/core';
import { Platform } from 'ionic-angular';
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';

import { HomePage } from '../pages/home/home';
import { SigninPage } from '../pages/signin/signin';

import { AngularFireAuth } from 'angularfire2/auth';

@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  rootPage: any;

  constructor(platform: Platform, statusBar: StatusBar, splashScreen: SplashScreen,
    afAuth: AngularFireAuth) {
    const authObserver = afAuth.authState.subscribe(user => {
      if (user) {
        this.rootPage = HomePage;
        authObserver.unsubscribe();
      } else {
        this.rootPage = SigninPage;
        authObserver.unsubscribe();
      }
    });

    platform.ready().then(() => {
```

```

    statusBar.styleDefault();
    splashScreen.hide();
  });
}
}

```

app.module.ts

```

import { BrowserModule } from '@angular/platform-browser';
import { ErrorHandler, NgModule } from '@angular/core';
import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';
import { SplashScreen } from '@ionic-native/splash-screen';
import { StatusBar } from '@ionic-native/status-bar';

import { AngularFireModule } from 'angularfire2';
import {    AngularFireDatabase,    AngularFireDatabaseModule    }    from
'angularfire2/database';
import { AngularFireAuthModule } from 'angularfire2/auth';

import { MyApp } from './app.component';
import { HomePage } from '../pages/home/home';
import { SigninPage } from '../pages/signin/signin';
import { SignupPage } from '../pages/signup/signup';
import { ResetpasswordPage } from '../pages/resetpassword/resetpassword';

import { EvControlPage } from '../pages/ev-control/ev-control';
import { EvAddPage } from '../pages/ev-add/ev-add';
import { EvAddPageModule } from '../pages/ev-add/ev-add.module';

import { CControlPage } from '../pages/c-control/c-control'
import { CAddPage } from '../pages/c-add/c-add';
import { CAddPageModule } from '../pages/c-add/c-add.module';

```

```

import { LControlPage } from '../pages/l-control/l-control';
import { LAddPage } from '../pages/l-add/l-add';
import { LAddPageModule } from '../pages/l-add/l-add.module';

import { AuthService } from '../providers/auth/auth-service';

import { EventosProvider } from '../providers/eventos/eventos';
import { CursosProvider } from '../providers/cursos/cursos';
import { LocaisProvider } from '../providers/locais/locais';

const firebaseConfig = {
  apiKey: "AlzaSyDhqijPVnyH4dHbP8YDKC4ZOM4L9_juCNk",
  authDomain: "uniinfo-database-full.firebaseio.com",
  databaseURL: "https://uniinfo-database-full.firebaseio.com",
  projectId: "uniinfo-database-full",
  storageBucket: "uniinfo-database-full.appspot.com",
  messagingSenderId: "613710825075"
};

@NgModule({
  declarations: [
    MyApp,
    HomePage,
    SigninPage,
    SignupPage,
    ResetpasswordPage,

    EvControlPage,
    CControlPage,
    LControlPage
  ],
  imports: [
    BrowserModule,

```

```

IonicModule.forRoot(MyApp, {
  platforms: {
    ios: {
      backButtonText: 'Voltar'
    }
  }
}),

EvAddPageModule,
CAddPageModule,
LAddPageModule,

AngularFireModule.initializeApp(firebaseConfig),
AngularFireAuthModule,
AngularFireDatabaseModule,

],
bootstrap: [IonicApp],
entryComponents: [
  MyApp,
  HomePage,
  SigninPage,
  SignupPage,
  ResetpasswordPage,

  CControlPage,
  EvControlPage,
  LControlPage,

  EvAddPage,
  CAddPage,
  LAddPage
],
providers: [

```

```

    StatusBar,
    SplashScreen,
    { provide: ErrorHandler, useClass: IonicErrorHandler },
    AuthService,
    EventosProvider,
    CursosProvider,
    LocaisProvider
  ]
})
export class AppModule { }

```

c-add.html

```

<ion-header>
  <ion-navbar>
    <div>
      <br>
      <br>
      <h1>
        <b>&nbsp; &nbsp; {{titulo}}</b>
      </h1>
      <font color="#3db7ff" size="2">
        <p align="left">&nbsp; &nbsp; &nbsp; &nbsp; Evento</p>
      </font>
    </div>
  </ion-navbar>
</ion-header>

<ion-content padding>

  <div class="div-mod-3">
    <form [formGroup]="form">

```

```

<ion-item>
  <ion-label>Turno</ion-label>
  <ion-select formControlName="turno">
    <ion-option value="Noturno">Noturno</ion-option>
    <ion-option value="Matiturno">Matiturno</ion-option>
  </ion-select>
</ion-item>

```

```

<ion-item>
  <ion-label floating>Curso</ion-label>
  <ion-input type="text" formControlName="curso"></ion-input>
</ion-item>

```

```

<ion-item *ngIf="!form.controls.curso.valid && (form.controls.curso.dirt ||
form.controls.curso.touched)" color="danger">
  <div [hidden]="!form.controls.curso.errors.required">
    O campo é obrigatório.
  </div>
</ion-item>

```

```

<ion-item>
  <ion-label floating>Semestre</ion-label>
  <ion-input type="number" formControlName="semestre"></ion-input>
</ion-item>

```

```

<ion-item *ngIf="!form.controls.semestre.valid && (form.controls.semestre.dirt ||
form.controls.semestre.touched)" color="danger">
  <div [hidden]="!form.controls.semestre.errors.required">
    O campo é obrigatório.
  </div>
</ion-item>

```

```

<ion-item>
  <ion-label floating>Segunda</ion-label>
  <ion-input type="text" formControlName="sseg"></ion-input>
</ion-item>

```



```

<!--

<ion-item>
  <ion-label floating>Terça</ion-label>
  <ion-input type="text" formControlName="ster"></ion-input>
</ion-item>

<ion-item>
  <ion-label floating>Quarta</ion-label>
  <ion-input type="text" formControlName="squa"></ion-input>
</ion-item>

<ion-item>
  <ion-label floating>Quinta</ion-label>
  <ion-input type="text" formControlName="squi"></ion-input>
</ion-item>

<ion-item>
  <ion-label floating>Sexta</ion-label>
  <ion-input type="text" formControlName="ssex"></ion-input>
</ion-item>
-->

  <div padding>
    <button      ion-button      block      type="submit"      [disabled]="!form.valid"
(click)="onSubmit()">Salvar</button>
  </div>
</form>
</div>

</ion-content>
c-add.ts
import { Component } from '@angular/core';

```

```
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';
```

```
import { CursosProvider } from '../providers/cursos/cursos';
```

```
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-c-add',
```

```
  templateUrl: 'c-add.html',
```

```
})
```

```
export class CAddPage {
```

```
  titulo: string;
```

```
  form: FormGroup;
```

```
  cursos: any;
```

```
  constructor(public navCtrl: NavController,
```

```
    public navCtrl: NavController,
```

```
    private formBuilder: FormBuilder,
```

```
    private provider: CursosProvider,
```

```
    private toast: ToastController) {
```

```
    this.cursos = this.navCtrl.data.cursos || {};
```

```
    this.createForm();
```

```
    this.setupPageTitle();
```

```
}
```

```
  private setupPageTitle() {
```

```
    this.titulo = this.navCtrl.data.cursos ? 'Alterando sala' : 'Nova sala';
```

```
}
```

```
  createForm() {
```

```

this.form = this.formBuilder.group({
  key: [this.cursos.key],
  turno: [this.cursos.turno, Validators.required],
  curso: [this.cursos.curso, Validators.required],
  semestre: [this.cursos.semestre, Validators.required],
  sseg: [this.cursos.sseg],

  /*
  ster: [this.cursos.ster],
  squa: [this.cursos.squa],
  squi: [this.cursos.squi],
  ssex: [this.cursos.ssex]
  */
})
}

onSubmit(){
  if (this.form.valid) {
    this.provider.save(this.form.value)
      .then(() => {
        this.toast.create({ message: 'Sala adicionada com sucesso.', duration:
3000}).present();
        //this.navCtrl.pop();
      })
      .catch((e) => {
        this.toast.create({ message: 'Erro ao salvar sala.', duration: 3000}).present();
        console.error(e);
      });
  }
}
}

```

c-control.html

<ion-header>

## <ion-navbar>

<div>

<br>

<br>

# 

**Registros**

# 

<font color="#3db7ff" size="2">

          Cursors

&lt;/font&gt;

&lt;ion-title&gt;

&lt;/ion-title&gt;

&lt;/div&gt;

&lt;/ion-navbar&gt;

&lt;/ion-header&gt;

<ion-content padding>

<div class="div-mod-control">

<br>

<br>

<ion-list>

```
<ion-list-header align="center">
```

# Cursos matiturnos

&lt;/ion-list-header&gt;

```
<div *ngFor="let curso of courses | async">
```

```
<ion-item *ngIf="curso.turno === 'Matiturno'">
```

<font color="#b2b2b2" size="2" align="center">

{{curso.curso}} | {{curso.semestre}}º Sem | Sala: {{curso.sseg}}

```

</font>

<button          ion-button          clear          item-start
(click)="editCurso(curso)">EDITAR</button>
<button          ion-button          clear          color="danger"          item-end
(click)="removeCurso(curso.key)">EXCLUIR</button>
</ion-item>
</div>
</ion-list>
<br>
<br>
<ion-list>
  <ion-list-header align="center">
    <h1>Cursos noturnos</h1>
  </ion-list-header>
  <div *ngFor="let curso of courses | async">
    <ion-item *ngIf="curso.turno === 'Noturno'">

      <font color="#b2b2b2" size="2" align="center">
        {{curso.curso}} | {{curso.semestre}}° Sem | Sala: {{curso.sseg}}
      </font>

      <button          ion-button          clear          item-start
(click)="editCurso(curso)">EDITAR</button>
      <button          ion-button          clear          color="danger"          item-end
(click)="removeCurso(curso.key)">EXCLUIR</button>
      </ion-item>
    </div>
  </ion-list>
  <br>
  <br>
  <font style="font-size:9pt; color:gray">
    <p align="right">Powered by UnilInfo.</p>
  </font>

```

```
</div>
```

```
</ion-content>
```

```
<ion-footer>
```

```
<font>
```

```
<p align="center">UnilInfo 2017 &#174; - Todos os direitos reservados.</p>
```

```
</font>
```

```
</ion-footer>
```

```
c-control.ts
```

```
import { Component } from '@angular/core';
```

```
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';
```

```
import { CursosProvider } from '../providers/cursos/cursos';
```

```
import { Observable } from 'rxjs/Observable';
```

```
import { CAddPage } from '../c-add/c-add';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-c-control',
```

```
  templateUrl: 'c-control.html',
```

```
})
```

```
export class CControlPage {
```

```

courses: Observable<any>;

constructor(public navCtrl: NavController,
  public navParams: NavParams,
  private provider: CursosProvider,
  private toast: ToastController) {

  this.courses = this.provider.getAll();
}

removeCurso(key: string) {
  if (key) {
    this.provider.remove(key)
      .then(() => {
        this.toast.create({ message: 'Registro removido com sucesso.', duration:
3000}).present();
      })
      .catch(() => {
        this.toast.create({ message: 'Erro ao remover registro.', duration:
3000}).present();
      });
  }
}

editCurso(cursos: any) {
  this.navCtrl.push('CAddPage', { cursos: cursos });
}
}

```

```

ev-add.html
<ion-header>

```

```

<ion-navbar>
  <div>
    <br>
    <br>
    <h1>
      <b>&nbsp; &nbsp; {{titulo}}</b>
    </h1>
    <font color="#3db7ff" size="2">
      <p align="left">&nbsp; &nbsp; &nbsp; &nbsp; Evento</p>
    </font>
  </div>
</ion-navbar>
</ion-header>

```

```

<ion-content padding>
<div class="div-mod-3">
  <form [formGroup]="form">
    <ion-item>
      <ion-label floating>Título</ion-label>
      <ion-input type="text" formControlName="titulo"></ion-input>
    </ion-item>
    <ion-item *ngIf="!form.controls.titulo.valid && (form.controls.titulo.dirt ||
form.controls.titulo.touched)" color="danger">
      <div [hidden]="!form.controls.titulo.errors.required">
        O campo é obrigatório.
      </div>
    </ion-item>

    <ion-item>
      <ion-label floating>Descrição</ion-label>
      <ion-input type="text" formControlName="descricao"></ion-input>
    </ion-item>

```



```

    <ion-item *ngIf="!form.controls.descricao.valid && (form.controls.descricao.dirt ||
form.controls.descricao.touched)" color="danger">
      <div [hidden]="!form.controls.descricao.errors.required">
        O campo é obrigatório.
      </div>
    </ion-item>

    <ion-item>
      <ion-label floating>Local</ion-label>
      <ion-input type="text" formControlName="local"></ion-input>
    </ion-item>
    <ion-item *ngIf="!form.controls.local.valid && (form.controls.local.dirt ||
form.controls.local.touched)" color="danger">
      <div [hidden]="!form.controls.local.errors.required">
        O campo é obrigatório.
      </div>
    </ion-item>

    <ion-item>
      <ion-label floating>Data</ion-label>
      <ion-input type="text" formControlName="data"></ion-input>
    </ion-item>
    <ion-item *ngIf="!form.controls.data.valid && (form.controls.data.dirt ||
form.controls.data.touched)" color="danger">
      <div [hidden]="!form.controls.data.errors.required">
        O campo é obrigatório.
      </div>
    </ion-item>

    <br>
    <br>
    <br>

    <div padding>

```

```

        <button      ion-button      block      type="submit"      [disabled]="!form.valid"
(click)="onSubmit()">Salvar</button>
    </div>
</form>
</div>
</ion-content>

```

ev-add.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';

```

```

import { EventosProvider } from '../providers/eventos/eventos';

```

```

import { FormBuilder, FormGroup, Validators } from '@angular/forms';

```

```

@IonicPage()
@Component({
  selector: 'page-ev-add',
  templateUrl: 'ev-add.html',
})

```

```

export class EvAddPage {
  titulo: string;
  form: FormGroup;
  evento: any;

```

```

  constructor(public navCtrl: NavController,
    public NavParams: NavParams,
    private FormBuilder: FormBuilder,
    private provider: EventosProvider,
    private toast: ToastController) {

```

```

this.evento = this.navParams.data.evento || {};
this.createForm();

this.setupPageTitle();
}

private setupPageTitle() {
  this.titulo = this.navParams.data.evento ? 'Alterando Evento' : 'Novo Evento';
}

createForm() {
  this.form = this.formBuilder.group({
    key: [this.evento.key],
    titulo: [this.evento.titulo, Validators.required],
    descricao: [this.evento.descricao, Validators.required],
    local: [this.evento.local, Validators.required],
    data: [this.evento.data, Validators.required]
  })
}

onSubmit(){
  if (this.form.valid) {
    this.provider.save(this.form.value)
      .then(() => {
        this.toast.create({ message: 'Evento salvo com sucesso.', duration:
3000}).present();
        this.navCtrl.pop();
      })
      .catch((e) => {
        this.toast.create({ message: 'Erro ao salvar contato.', duration: 3000}).present();
        console.error(e);
      });
  }
}

```

}

ev-control.html

<ion-header>

## <ion-navbar>

<div>

<br>

<br>

# <h1>

**Registros**

# 

<font color="#3db7ff" size="2">

          Eventos

&lt;/font&gt;

&lt;ion-title&gt;

&lt;/ion-title&gt;

&lt;/ion-navbar&gt;

&lt;/ion-header&gt;

<ion-content padding>

<div class="div-mod-control">

<br>

<br>

&lt;ion-list&gt;

```
<ion-list-header align="center">
```

# Eventos registrados

&lt;/ion-list-header&gt;

```
<ion-item *ngFor="let evento of events | async">
```

<font color="#b2b2b2" size="2" align="center">

```

        {{evento.titulo}}
    </font>
    <button          ion-button          clear          item-start
(click)="editEvento(evento)">EDITAR</button>
    <button          ion-button          clear          color="danger"          item-end
(click)="removeEvento(evento.key)">EXCLUIR</button>
    </ion-item>
    <br>
    <br>
    <font style="font-size:9pt; color:gray">
        <p align="right">Powered by UniInfo.</p>
    </font>
</ion-list>

</div>

```

```

</ion-content>

```

```

<ion-footer>
    <font>
        <p align="center">UniInfo 2017 &#174; - Todos os direitos reservados.</p>
    </font>
</ion-footer>

```

```

ev-control.ts

```

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';

import { EventosProvider } from '../providers/eventos/eventos';

import { Observable } from 'rxjs/Observable';

```

```

import { EvAddPage } from '../ev-add/ev-add';

@IonicPage()
@Component({
  selector: 'page-ev-control',
  templateUrl: 'ev-control.html',
})
export class EvControlPage {

  events: Observable<any>;

  constructor(public navCtrl: NavController,
    private provider: EventosProvider,
    private toast: ToastController) {

    this.events = this.provider.getAll();
  }

  removeEvento(key: string) {
    if (key) {
      this.provider.remove(key)
        .then(() => {
          this.toast.create({ message: 'Evento removido com sucesso.', duration: 3000
        }).present();
        })
        .catch(() => {
          this.toast.create({ message: 'Erro ao remover o evento.', duration: 3000
        }).present();
        });
    }
  }

  editEvento(evento: any) {

```

```

    this.navCtrl.push('EvAddPage', { evento: evento });
  }

}

```

home.html

```

<ion-header>
  <ion-navbar>
    <div>
      
      <p align="center">Ambiente administrativo</p>
    </div>
  </ion-navbar>

```

```

</ion-header>

```

```

<ion-content padding>
  <div class="div-center-home">

```

```

    <!--Eventos-->

```

```

    <br>

```

```

    <br>

```

```

  <div class="div-mod-home">
    <ion-card class="card-mod-home">
      

      <ion-card-content>

```

```

<ion-card-title>
  <ion-item>
    <ion-icon name="ios-calendar" color="primary" item-start>
    </ion-icon>
    <h2>Eventos</h2>
    <h3>Criar, editar ou excluir registros de eventos.</h3>
    <br>
    <p>A aba eventos possui os registros de
      <br>avisos importantes ao aluno referente
      <br>a universsidade. O que deseja fazer? <br><br>
    </p>
  </ion-item>
</ion-card-title>

<ion-item>
  <ion-icon name="ios-eye" item-start></ion-icon>
  <button ion-button clear (click)="goToEvControl()">
    Ir para eventos
  </button>
</ion-item>

<ion-item>
  <ion-icon name="ios-add" item-start></ion-icon>
  <button ion-button clear (click)="goToEvAdd()">
    Adicionar evento
  </button>
</ion-item>

</ion-card-content>
</ion-card>
</div>

```



```
<!--Salas-->
```

```
<div class="div-mod-home">
```

```
<ion-card class="card-mod-home">
```

```

```

```
<ion-card-content>
```

```
<ion-card-title>
```

```
<ion-item>
```

```
<ion-icon name="ios-school" color="primary" item-start>
```

```
</ion-icon>
```

```
<h2>Cursos</h2>
```

```
<h3>Criar, editar ou excluir registros de salas.</h3>
```

```
<br>
```

```
<p>A aba cursos possui os registros de
```

```
<br>todas as salas, por dia, de cada curso.
```

```
<br>O que deseja fazer? <br><br>
```

```
</p>
```

```
</ion-item>
```

```
</ion-card-title>
```

```
<ion-item>
```

```
<ion-icon name="ios-eye" item-start></ion-icon>
```

```
<button ion-button clear (click)="goToCCControl()">
```

```
  Ir para salas
```

```
</button>
```

```
</ion-item>
```

```
<ion-item>
```

```
  <ion-icon name="ios-add" item-start></ion-icon>
```

```
  <button ion-button clear (click)="goToCAdd()">
```

```
    Adicionar sala
```

```
  </button>
```

```
</ion-item>
```

```
</ion-card-content>
```

```
</ion-card>
```

```
</div>
```

```
<!--Locais-->
```

```
<div class="div-mod-home">
```

```
<ion-card class="card-mod-home">
```

```
  
```

```
<ion-card-content>
```

```
<ion-card-title>
```

```
  <ion-item>
```

```
    <ion-icon name="ios-pin" color="primary" item-start>
```

```
  </ion-icon>
```

```
  <h2>Locais</h2>
```

```
  <h3>Criar, editar ou excluir registros de locais.</h3>
```

```

    <br>
    <p>A aba locais possui os registros de
      <br>todos os pontos de referências do
      <br>campus. O que deseja fazer? <br><br>
    </p>
  </ion-item>

</ion-card-title>

<ion-item>
  <ion-icon name="ios-eye" item-start></ion-icon>
  <button ion-button clear (click)="goToLControl()">
    Ir para locais
  </button>
</ion-item>
<ion-item>
  <ion-icon name="ios-add" item-start></ion-icon>
  <button ion-button clear (click)="goToLAdd()">
    Adicionar local
  </button>
</ion-item>

</ion-card-content>
</ion-card>
</div>

```

```

<!--Mais-->

```

```

<div class="div-mod-home">
<ion-card class="card-mod-home">
  

  <ion-card-content>

    <ion-card-title>
      <ion-item>
        <ion-icon name="ios-contacts" color="primary" item-start>
        </ion-icon>
        <h2>Usuários</h2>
        <h3>Ações de usuários.</h3>
        <br>
        <p>
          <b>Atenção</b>! Usuários administradores tem
          <br> acesso completo ao sistema, assim como
          <br> todas as permissões de alteração do
          <br> banco de dados.</p>
        </ion-item>
      </ion-card-title>
      <ion-list>
        <ion-item>
          <ion-icon name="add" item-start></ion-icon>
          <button ion-button clear (click)="goToSignUp()">
            Adicionar administrador
          </button>
        </ion-item>
        <ion-item>
          <ion-icon name="ios-exit" item-start></ion-icon>
          <button ion-button clear (click)="signOut()">
            Fazer logoff
          </button>
        </ion-item>
      </ion-list>
    </ion-card-content>
  </ion-card>
</div>

```

```
</ion-list>
```

```
</ion-card-content>
```

```
</ion-card>
```

```
</div>
```

```
</div>
```

```
</ion-content>
```

```
<ion-footer>
```

```
<font>
```

```
<p align="center">UnilInfo 2017 &#174; - Todos os direitos reservados.</p>
```

```
</font>
```

```
</ion-footer>
```

```
home.ts
```

```
import { Component } from '@angular/core';
```

```
import { NavController } from 'ionic-angular';
```

```
import { AuthService } from '../providers/auth/auth-service';
```

```
import { SigninPage } from '../signin/signin';
```

```
import { SignupPage } from '../signup/signup';
```

```
import { EvAddPage } from '../ev-add/ev-add';
```

```
import { EvControlPage } from '../ev-control/ev-control';
```

```
import { CAddPage } from '../c-add/c-add';
```

```
import { CControlPage } from '../c-control/c-control';
```

```
import { LAddPage } from '../l-add/l-add';
```

```
import { LControlPage } from '../l-control/l-control';
```

```
@Component({
  selector: 'page-home',
  templateUrl: 'home.html'
})
export class HomePage {

  constructor(public navCtrl: NavController, private authService: AuthService) {

  }

  goToEvControl(){
    this.navCtrl.push(EvControlPage)
  }

  goToEvAdd(){
    this.navCtrl.push(EvAddPage)
  }

  goToCControl(){
    this.navCtrl.push(CControlPage)
  }

  goToCAdd(){
    this.navCtrl.push(CAddPage)
  }

  goToLControl(){
    this.navCtrl.push(LControlPage)
  }

  goToLAdd(){
    this.navCtrl.push(LAddPage)
  }
}
```

```

goToSignUp(){
  this.navCtrl.push(SignupPage)
}

signOut() {
  this.authService.signOut()
    .then(() => {
      this.navCtrl.setRoot(SigninPage);
    })
    .catch((error) => {
      console.error(error);
    });
}
}

```

l-add.html

```

<ion-header>
  <ion-navbar>
    <div>
      <br>
      <br>
      <h1>
        <b>&nbsp; &nbsp; {{titulo}}</b>
      </h1>
      <font color="#3db7ff" size="2">
        <p align="left">&nbsp; &nbsp; &nbsp; &nbsp; Local</p>
      </font>
    </div>
  </ion-navbar>

```

```
</ion-header>
```

```
<ion-content padding>
```

```
<div class="div-mod-3">
```

```
<form [formGroup]="form">
```

```
<ion-item>
```

```
<ion-label floating>Ambiente</ion-label>
```

```
<ion-input type="text" formControlName="ambiente"></ion-input>
```

```
</ion-item>
```

```
<ion-item *ngIf="!form.controls.ambiente.valid && (form.controls.ambiente.dirt ||
form.controls.ambiente.touched)" color="danger">
```

```
<div [hidden]="!form.controls.ambiente.errors.required">
```

```
  O campo é obrigatório.
```

```
</div>
```

```
</ion-item>
```

```
<ion-item>
```

```
<ion-label floating>Localização</ion-label>
```

```
<ion-input type="text" formControlName="localizacao"></ion-input>
```

```
</ion-item>
```

```
<ion-item *ngIf="!form.controls.localizacao.valid &&
(form.controls.localizacao.dirt || form.controls.localizacao.touched)" color="danger">
```

```
<div [hidden]="!form.controls.localizacao.errors.required">
```

```
  O campo é obrigatório.
```

```
</div>
```

```
</ion-item>
```

```
<ion-item>
```

```
<ion-label floating>Horário</ion-label>
```

```
<ion-input type="text" formControlName="horario"></ion-input>
```

```
</ion-item>
```

```
<ion-item *ngIf="!form.controls.horario.valid && (form.controls.horario.dirt ||
form.controls.horario.touched)" color="danger">
```

```
<div [hidden]="!form.controls.horario.errors.required">
```



```

    O campo é obrigatório.
  </div>
</ion-item>

<ion-item>
  <ion-label floating>Descrição</ion-label>
  <ion-input type="text" formControlName="descricao"></ion-input>
</ion-item>
<ion-item
  *ngIf="!form.controls.descricao.valid"
  (form.controls.descricao.dirt || form.controls.descricao.touched)" color="danger">
  <div [hidden]="!form.controls.descricao.errors.required">
    O campo é obrigatório.
  </div>
</ion-item>

<div padding>
  <button ion-button block type="submit" [disabled]="!form.valid"
    (click)="onSubmit()">Salvar</button>
</div>
</form>
</div>
</ion-content>

```

l-add.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';

import { LocaisProvider } from '../providers/locais/locais';

import { FormBuilder, FormGroup, Validators } from '@angular/forms';

@IonicPage()
@Component({

```

```

    selector: 'page-l-add',
    templateUrl: 'l-add.html',
  })
export class LAddPage {
  titulo: string;
  form: FormGroup;
  local: any;

  constructor(public navCtrl: NavController,
    public navParams: NavParams,
    private FormBuilder: FormBuilder,
    private provider: LocaisProvider,
    private toast: ToastController) {

    this.local = this.navParams.data.local || {};
    this.createForm();

    this.setupPageTitle();
  }

  private setupPageTitle() {
    this.titulo = this.navParams.data.local ? 'Alterando Local' : 'Novo Local';
  }

  createForm() {
    this.form = this.formBuilder.group({
      key: [this.local.key],
      ambiente: [this.local.ambiente, Validators.required],
      localizacao: [this.local.localizacao, Validators.required],
      horario: [this.local.horario, Validators.required],
      descricao: [this.local.descricao, Validators.required]
    })
  }
}

```



```
</ion-title>
```

```
</div>
```

```
</ion-navbar>
```

```
</ion-header>
```

```
<ion-content padding>
```

```
<div class="div-mod-control">
```

```
<br>
```

```
<br>
```

```
<ion-list>
```

```
<ion-list-header align="center">
```

```
<h1>Locais registrados</h1>
```

```
</ion-list-header>
```

```
<ion-item *ngFor="let local of places | async">
```

```
<font color="#b2b2b2" size="2" align="center">
```

```
{{local.ambiente}} | {{local.localizacao}}
```

```
</font>
```

```
<button ion-button clear item-start (click)="editLocal(local)">EDITAR</button>
```

```
<button ion-button clear color="danger" item-end
```

```
(click)="removeLocal(local.key)">EXCLUIR</button>
```

```
</ion-item>
```

```
<br>
```

```
<br>
```

```
<font style="font-size:9pt; color:gray">
```

```
<p align="right">Powered by UniInfo.</p>
```

```
</font>
```

```
</ion-list>
```

```
</div>
```

```
</ion-content>
```

```

<ion-footer>
  <font>
    <p align="center">UnilInfo 2017 &#174; - Todos os direitos reservados.</p>
  </font>
</ion-footer>

```

l-control.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';

import { LocaisProvider } from '../providers/locais/locais';

import { Observable } from 'rxjs/Observable';

import { LAddPage } from '../l-add/l-add';

@IonicPage()
@Component({
  selector: 'page-l-control',
  templateUrl: 'l-control.html',
})
export class LControlPage {

  places: Observable<any>;

  constructor(public navCtrl: NavController,
    public NavParams: NavParams,
    private provider: LocaisProvider,
    private toast: ToastController) {

    this.places = this.provider.getAll();
  }

```

```
}
```

```
removeLocal(key: string) {
  if (key) {
    this.provider.remove(key)
    .then(() => {
      this.toast.create({ message: 'Ambiente removido com sucesso.', duration:
3000}).present();
    })
    .catch(() => {
      this.toast.create({ message: 'Erro ao remover ambiente.', duration:
3000}).present();
    });
  }
}
```

```
editLocal(local: any) {
  this.navCtrl.push('LAddPage', { local: local });
}

}
```

resetpassword.html

```
<ion-header>
<ion-navbar>
  <div>
    <br>
    <br>
    <h1>
      <b>&nbsp; &nbsp; Recuperação de senha</b>
    </h1>
```



```
</ion-content>
```

```
<ion-footer>
```

```
  <font>
```

```
    <p align="center">UnilInfo 2017 &#174; - Todos os direitos reservados.</p>
```

```
  </font>
```

```
</ion-footer>
```

```
resetpassword.ts
```

```
import { Component, ViewChild } from '@angular/core';
```

```
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';
```

```
import { NgForm } from '@angular/forms';
```

```
import { AuthService } from '../providers/auth/auth-service';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-resetpassword',
```

```
  templateUrl: 'resetpassword.html',
```

```
})
```

```
export class ResetpasswordPage {
```

```
  userEmail: string = '';
```

```
  @ViewChild('form') form: NgForm;
```

```
  constructor(public navCtrl: NavController,
```

```
               public navCtrl: NavParams,
```

```
               private toastCtrl: ToastController,
```

```
               private authService: AuthService) {
```

```
  }
```

```
  resetPassword() {
```

```
    if (this.form.form.valid){
```



```

let toast = this.toastCtrl.create({duration: 3000, position: 'bottom'});
this.authService.resetPassword(this.userEmail)
.then(() => {
  toast.setMessage('Sua solicitação foi enviada para seu e-mail.')
  toast.present()

  this.navCtrl.pop();
})
.catch((error: any) => {

  if (error.code == 'auth/invalid-email') {
    toast.setMessage('E-mail inválido.')
  }
  else if (error.code == 'auth/user-not-found') {
    toast.setMessage('Usuário não encontrado.');
```

signin.html

```
<ion-content padding>
```

```
<form #form="ngForm" novalidate>
```

```
<div class="center-div">
```

```
<font color="#7c7c7c" size="2">
```

```
<p align="center">Ambiente administrativo</p>
```

```
<br>
```

```
<br>
</font>
```

```

```

```
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
```

```
<ion-list>
```

```
<ion-item>
```

```
<ion-label floating>E-mail</ion-label>
```

```
<ion-input type="email" name="email" [(ngModel)]="user.email"
#email="ngModel" required></ion-input>
```

```
</ion-item>
```

```
<ion-item *ngIf="email.errors && (email.dirty || email.touched)" class="text-
danger">
```

```
O campo é obrigatório.
```

```
</ion-item>
```

```
<ion-item>
```

```
<ion-label floating>Senha</ion-label>
```

```
<ion-input type="password" name="password" [(ngModel)]="user.password"
#password="ngModel" required></ion-input>
```

```
</ion-item>
```

```

        <ion-item *ngIf="password.errors && (password.dirty || password.touched)"
class="text-danger">
            O campo é obrigatório
        </ion-item>
    </ion-list>

    <div align="center">

        <br>
        <br>

        <button      ion-button      color="primary"      [disabled]="!form.form.valid"
(click)="signIn()">Entrar</button>

        <br>
        <br>

        <button ion-button clear full (click)="goResetpassword()">ESQUECI MINHA
SENHA</button>

    </div>
</div>
</form>

</ion-content>

<ion-footer>
    <font>
        <p align="center">UnilInfo 2017 &#174; - Todos os direitos reservados.</p>
    </font>
</ion-footer>

```

signin.ts

```
import { Component, ViewChild } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';
```

```
import { NgForm } from '@angular/forms';
import { User } from '../providers/auth/user';
import { AuthService } from '../providers/auth/auth-service';
```

```
import { ResetpasswordPage } from '../resetpassword/resetpassword';
import { HomePage } from '../home/home';
```

```
@IonicPage()
```

```
@Component({
  selector: 'page-signin',
  templateUrl: 'signin.html',
})
```

```
export class SigninPage {
  user: User = new User();
  @ViewChild('form') form: NgForm;
```

```
  constructor(public navCtrl: NavController,
               public navCtrl: NavParams,
               private toastCtrl: ToastController,
               private authService: AuthService) {
  }
```

```
  goResetpassword() {
    this.navCtrl.push(ResetpasswordPage)
  }
```

```
  signIn() {
    if (this.form.valid) {
      this.authService.signIn(this.user)
        .then(() => {
```

```

    this.navCtrl.setRoot(HomePage)
  })
  .catch((error: any) => {
    let toast = this.toastCtrl.create({ duration:3000, position: 'bottom' });

    if (error.code == 'auth/invalid-email') {
      toast.setMessage('E-mail informado é inválido.');
```

```

    }

    else if (error.code == 'auth/user-disabled') {
      toast.setMessage('Usuário está desativado.');
```

```

    }

    else if (error.code == 'auth/user-not-found') {
      toast.setMessage('Usuário não foi encontrado.');
```

```

    }

    else if (error.code == 'auth/wrong-password') {
      toast.setMessage('A senha digitada não é válida.');
```

```

    }
    toast.present();
  });
}
}
}

```

signup.html

<ion-header>

<ion-navbar>

<ion-title>Criar uma nova conta</ion-title>

```
</ion-navbar>
```

```
</ion-header>
```

```
<ion-content padding>
```

```
<form #form="ngForm" novalidate>
```

```
<ion-list>
```

```
<ion-item>
```

```
<ion-label stacked>E-mail</ion-label>
```

```
<ion-input type="text" name="email" [(ngModel)]="user.email"
```

```
#email="ngModel" required></ion-input>
```

```
</ion-item>
```

```
<ion-item *ngIf="email.errors && (email.dirty || email.touched)" class="text-
danger">
```

```
    O campo é obrigatório
```

```
</ion-item>
```

```
<ion-item>
```

```
<ion-label stacked>Senha</ion-label>
```

```
<ion-input type="password" name="password" [(ngModel)]="user.password"
```

```
#password="ngModel" require></ion-input>
```

```
</ion-item>
```

```
<ion-item *ngIf="password.errors && (password.dirty || password.touched)"
class="text-danger">
```

```
    O campo é obrigatório
```

```
</ion-item>
```

```
</ion-list>
```

```
<button ion-button block color="primary" [disabled]="!form.form.valid"
(click)="createAccount()">
```

```
    CRIAR CONTA
```

```
</button>
```

```
</form>
```

```
</ion-content>
```

```
signup.ts
```

```
//Importações padrões do Ionic
```

```
import { Component, ViewChild } from '@angular/core';
```

```
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';
```

```
//Importações do provedor
```

```
import { NgForm } from '@angular/forms';
```

```
import { User } from '../providers/auth/user';
```

```
import { AuthService } from '../providers/auth/auth-service';
```

```
//Importação de página
```

```
import { HomePage } from '../home/home';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-signup',
```

```
  templateUrl: 'signup.html',
```

```
})
```

```
export class SignupPage {
```

```
  user: User = new User();
```

```
  @ViewChild('form') form: NgForm;
```

```
  constructor(
```

```
    public navCtrl: NavController,
```

```
    public NavParams: NavParams,
```

```
    public toastCtrl: ToastController,
```

```

public authService: AuthService) {
}

createAccount() {
  if (this.form.valid) {
    let toast = this.toastCtrl.create({ duration: 3000, position: 'bottom' });

    this.authService.createUser(this.user)
      .then((user: any) => {
        user.sendEmailVerification();

        toast.setMessage('Usuário cadastrado com sucesso.')
        toast.present();

        this.navCtrl.setRoot(HomePage)
      })
      .catch((error: any) => {
        if (error.code == 'auth/email-already-in-use') {
          toast.setMessage('O e-mail digitado já está em uso');
        } else if (error.code == 'auth/invalid-email') {
          toast.setMessage('O e-mail digitado não é válido');
        }
        else if (error.code == 'auth/operation-not-allowed') {
          toast.setMessage('Não esta habilitado criar usuários');
        }
        else if (error.code == 'auth/weak-password') {
          toast.setMessage('A senha informada é muito fraca');
        }
        toast.present();
      });
  }
}
}

```



Provider de AUTENTICAÇÃO auth-service.ts

```
import { Injectable } from '@angular/core';
```

```
import { Observable } from 'rxjs/Observable';
```

```
import { AngularFireAuth } from 'angularfire2/auth';
```

```
import { User } from './user';
```

```
import * as firebase from 'firebase/app';
```

```
@Injectable()
```

```
export class AuthService {
```

```
  user: Observable<firebase.User>;
```

```
  constructor(private angularFireAuth: AngularFireAuth) {
```

```
    this.user = angularFireAuth.authState;
```

```
  }
```

```
  signIn(user: User) {
```

```
    return this.angularFireAuth.auth.signInWithEmailAndPassword(user.email,  
user.password);
```

```
  }
```

```
  signOut() {
```

```
    return this.angularFireAuth.auth.signOut();
```

```
  }
```

```

resetPassword(email: string) {
  return this.angularFireAuth.auth.sendPasswordResetEmail(email);
}

createUser(user: User) {
  return this.angularFireAuth.auth.createUserWithEmailAndPassword(user.email,
user.password);
}
}

```

Provider de AUTENTICAÇÃO user.ts

```

export class User {
  email: string;
  password: string;
}

```

Provider cursors.ts

```

import { Injectable } from '@angular/core';
import { AngularFireDatabase, snapshotChanges } from 'angularfire2/database';

```

@Injectable()

```

export class CursosProvider {

  private PATH = 'c_data/';

  constructor(private db: AngularFireDatabase) { }

  getAll() {
    return this.db.list(this.PATH)
      .snapshotChanges()
      .map(changes => {
        return changes.map(c => ({

```

```

        key: c.payload.key,
        ...c.payload.val()
    ));
  })
}

get(key: string) {
  return this.db.object(this.PATH + key)
    .snapshotChanges()
    .map(c => {
      return { key: c.key, ...c.payload.val() };
    })
}

save(cursos: any) {
  return new Promise((resolve, reject) => {
    if (cursos.key) {
      this.db.list(this.PATH)
        .update(cursos.key, {
          turno: cursos.turno,
          curso: cursos.curso,
          semestre: cursos.semestre,
          sseg: cursos.sseg,
          /*
          ster: cursos.ster,
          squa: cursos.squa,
          squi: cursos.squi,
          ssex: cursos.ssex
          */
        })
        .then(() => resolve())
        .catch((e) => reject(e));
    } else {
      this.db.list(this.PATH)

```

```

        .push({
            turno: cursos.turno,
            curso: cursos.curso,
            semestre: cursos.semestre,
            sseg: cursos.sseg,
            /*
            ster: cursos.ster,
            squa: cursos.squa,
            squi: cursos.squi,
            ssex: cursos.ssex
            */
        })
        .then(() => resolve());
    }
});
}

remove(key: string) {
    return this.db.list(this.PATH).remove(key);
}

}

```

Provider eventos.ts

```

import { Injectable } from '@angular/core';
import { AngularFireDatabase, snapshotChanges } from 'angularfire2/database';

```

```
@Injectable()
```

```
export class EventosProvider {
```

```
    private PATH = 'e_data/';
```

```

constructor(private db: AngularFireDatabase) { }

getAll() {
  return this.db.list(this.PATH)
    .snapshotChanges()
    .map(changes => {
      return changes.map(ev => ({
        key: ev.payload.key,
        ...ev.payload.val()
      }));
    })
}

get(key: string) {
  return this.db.object(this.PATH + key)
    .snapshotChanges()
    .map( ev => {
      return { key: ev.key, ...ev.payload.val() };
    })
}

save(evento: any) {
  return new Promise((resolve, reject) => {
    if (evento.key) {
      this.db.list(this.PATH)
        .update(evento.key, { titulo: evento.titulo, descricao: evento.descricao,
local: evento.local, data: evento.data })
        .then(() => resolve())
        .catch((e) => reject(e));
    } else {
      this.db.list(this.PATH)
        .push({ titulo: evento.titulo, descricao: evento.descricao, local:
evento.local, data: evento.data })

```

```

        .then(() => resolve());
    }
});
}

remove(key: string) {
    return this.db.list(this.PATH).remove(key);
}

}

```

Provider locais.ts

```

import { Injectable } from '@angular/core';
import { AngularFireDatabase, snapshotChanges } from 'angularfire2/database';

```

@Injectable()

```

export class LocaisProvider {

```

```

    private PATH = 'l_data/';

```

```

    constructor(private db: AngularFireDatabase) { }

```

```

    getAll() {
        return this.db.list(this.PATH)
            .snapshotChanges()
            .map(changes => {
                return changes.map(l => ({
                    key: l.payload.key,
                    ...l.payload.val()
                }));
            })
    }
}

```

```

get(key: string) {
  return this.db.object(this.PATH + key)
    .snapshotChanges()
    .map(l => {
      return { key: l.key, ...l.payload.val() };
    })
}

save(local: any) {
  return new Promise((resolve, reject) => {
    if (local.key) {
      this.db.list(this.PATH)
        .update(local.key, { ambiente: local.ambiente, localizacao:
local.localizacao, horario: local.horario, descricao: local.descricao })
        .then(() => resolve())
        .catch((e) => reject(e));
    } else {
      this.db.list(this.PATH)
        .push({ ambiente: local.ambiente, localizacao: local.localizacao, horario:
local.horario, descricao: local.descricao })
        .then(() => resolve());
    }
  });
}

remove(key: string) {
  return this.db.list(this.PATH).remove(key);
}
}

```





## APÊNDICE E – CÓDIGO FONTE APLICATIVO MOBILE

```

app.component.ts
import { Component } from '@angular/core';
import { Platform } from 'ionic-angular';
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';

import { GlobalPreferencesVar } from '../providers/global_var/preferencias';
import { OfflineData } from '../providers/global_var/offline_data';

import { TabsPage } from '../pages/tabs/tabs';
import { StartPage } from '../pages/start/start';

@Component({
  templateUrl: 'app.html'
})
//DEFINE A PÁGINA DE ROOT
export class MyApp {

  rootPage: any;

  constructor(platform: Platform,
    statusBar: StatusBar,
    splashScreen: SplashScreen,
    private providerPreferences: GlobalPreferencesVar,
    private offlineProvider: OfflineData) {

    this.rootPage = this.offlineProvider.Status;

    platform.ready().then(() => {
      statusBar.styleDefault();
      splashScreen.hide();
    });
  }

```

```

}
}

```

app.module.ts

```
/* INSTALAÇÃO DOS MÓDULOS DO FIREBASE
```

```
  npm install firebase angularfire2 --save
```

```
*/
```

```
// IMPORTAÇÕES PADRÕES DO SISTEMA
```

```
import { NgModule, ErrorHandler } from '@angular/core';
```

```
import { BrowserModule } from '@angular/platform-browser';
```

```
import { IonicApp, IonicModule, IonicErrorHandler } from 'ionic-angular';
```

```
import { MyApp } from './app.component';
```

```
// IMPORTAÇÃO DAS PÁGINAS
```

```
import { HelpPage } from '../pages/help/help';
```

```
import { HelpPageModule } from '../pages/help/help.module';
```

```
import { EventosPage } from '../pages/eventos/eventos';
```

```
import { EventosPageModule } from '../pages/eventos/eventos.module';
```

```
import { CursosPage } from '../pages/cursos/cursos';
```

```
import { CursosPageModule } from '../pages/cursos/cursos.module';
```

```
import { CinfoPage } from '../pages/cinfo/cinfo';
```

```
import { CinfoPageModule } from '../pages/cinfo/cinfo.module';
```

```
import { CmorePage } from '../pages/cmore/cmore';
```

```
import { LocaisPage } from '../pages/locais/locais';
import { LocaisPageModule } from '../pages/locais/locais.module';
import { LinfoPage } from '../pages/linfo/linfo';
import { LinfoPageModule } from '../pages/linfo/linfo.module';

import { InfoPage } from '../pages/info/info';

// IMPORTAÇÃO DA PÁGINA DE PREFERÊNCIAS
import { StartPage } from '../pages/start/start';
import { StartPageModule } from '../pages/start/start.module';

// IMPORTAÇÃO DA PÁGINA DE ESCOLHA DE CURSO
import { SetCursoPage } from '../pages/set-curso/set-curso';
import { SetCursoPageModule } from '../pages/set-curso/set-curso.module';

import { SetCurso2Page } from '../pages/set-curso2/set-curso2';
import { SetCurso2PageModule } from '../pages/set-curso2/set-curso2.module';

import { SetCurso3Page } from '../pages/set-curso3/set-curso3';
import { SetCurso3PageModule } from '../pages/set-curso3/set-curso3.module';

// IMPORTAÇÃO DA PÁGINA DE CONFIRMAÇÃO
import { ConfirmPage } from '../pages/confirm/confirm';

// IMPORTAÇÃO DA BARRA INFERIOR
import { TabsPage } from '../pages/tabs/tabs';

// IMPORTAÇÃO DA SPLASH SCREEN E DA STATUS BAR
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';

// IMPORTAÇÃO DO MÓDULO DO BANCO DE DADOS
import { AngularFireModule } from 'angularfire2';
```

```
import { AngularFireDatabase, AngularFireDatabaseModule } from
'angularfire2/database';
```

```
// IDENTIFICAÇÃO EXCLUSIVA DO BANCO DE DADOS
```

```
const firebaseConfig = {
  apiKey: "AlzaSyDhqijPVnyH4dHbP8YDKC4ZOM4L9_juCNk",
  authDomain: "uniinfo-database-full.firebaseio.com",
  databaseURL: "https://uniinfo-database-full.firebaseio.com",
  projectId: "uniinfo-database-full",
  storageBucket: "uniinfo-database-full.appspot.com",
  messagingSenderId: "613710825075"
};
```

```
// IMPORTAÇÃO DO PROVIDER DE DADOS OFFLINE
```

```
import { OfflineData } from '../providers/global_var/offline_data';
import { DatePipe } from '@angular/common';
import { IonicStorageModule } from '@ionic/storage';
```

```
// IMPORTAÇÃO DOS PROVIDERS DO BANCO DE DADOS
```

```
import { CursosProvider } from '../providers/cursos/cursos';
import { EventosProvider } from '../providers/eventos/eventos';
import { LocaisProvider } from '../providers/locais/locais';
```

```
// IMPORTAÇÃO DOS PROVIDER DE VARIÁVEIS GLOBAIS
```

```
import { GlobalLocalVar } from '../providers/global_var/local';
import { GlobalCursoVar } from '../providers/global_var/curso';
```

```
// IMPORTAÇÃO DO PROVIDER DE VARIÁVEIS GLOBAIS DE PREFERÊNCIAS
```

```
import { GlobalPreferencesVar } from '../providers/global_var/preferencias';
import { ConfirmPageModule } from '../pages/confirm/confirm.module';
```

```
// DECLARAÇÃO DOS RECURSOS E PÁGINAS
```

```
@NgModule({
  declarations: [
```

```

MyApp,

InfoPage,
CmorePage,

TabsPage
],
imports: [
  BrowserModule,
  IonicModule.forRoot(MyApp, {
    platforms: {
      ios: {
        backButtonText: "
      }
    }
  }),
  IonicStorageModule.forRoot(),
  // PÁGINAS ACESSADAS POR MAIS DE UM EVENTO

  HelpPageModule,

  CinfoPageModule,
  LinfoPageModule,

  CursosPageModule,

  StartPageModule,
  SetCursoPageModule,
  SetCurso2PageModule,
  SetCurso3PageModule,
  ConfirmPageModule,
  EventosPageModule,
  LocaisPageModule,

```

```
AngularFireModule.initializeApp(firebaseConfig),
AngularFireDatabaseModule,
],
bootstrap: [IonicApp],
entryComponents: [
  MyApp,

  TabsPage,

  HelpPage,

  StartPage,
  SetCursoPage,
  SetCurso2Page,
  SetCurso3Page,
  ConfirmPage,

  EventosPage,

  CursosPage,
  CinfoPage,
  CmorePage,

  InfoPage,
  LinfoPage
],

// DECLARAÇÃO DOS PROVIDERS
providers: [
  StatusBar,
  SplashScreen,
  {provide: ErrorHandler, useClass: IonicErrorHandler},

  CursosProvider,
```

```

    EventosProvider,
    LocaisProvider,

    GlobalLocalVar,
    GlobalCursoVar,
    GlobalPreferencesVar,

    OfflineData,
    DatePipe
  ]
})
export class AppModule {

}

```

```

cinfo.html
<ion-header>
  <ion-navbar>
    <div>
      <br>
      <br>
      <h1>
        <b>&nbsp; &nbsp; &nbsp; Consulta Rápida</b>
      </h1>
      <font color="#3db7ff" size="2" align="left">

```





```

    <font color="#b2b2b2" size="2" align="left">
      <p align="left">* Turno:</p>
    </font>
  </ion-label>
  <ion-select      cancelText="Fechar"      okText="Pronto"      class='largura'
disabled="{{isEnabled1}}">
    <ion-option (ionSelect)="proc1('Matiturno')">Matiturno</ion-option>
    <ion-option (ionSelect)="proc1('Noturno')">Noturno</ion-option>
  </ion-select>
</ion-item>
<ion-item>
  <ion-label>
    <font color="#b2b2b2" size="2" align="left">
      <p align="left">* Curso:</p>
    </font>
  </ion-label>
  <ion-select      cancelText="Fechar"      okText="Pronto"      class='largura'
disabled="{{isEnabled2}}">
    <div *ngFor="let curso of courses | async">
      <ion-option *ngIf="curso.turno === providerPreferences.moreTurno &&
curso.semestre === '1'" (ionSelect)="proc2(curso.curso)">{{curso.curso}}</ion-
option>
    </div>
  </ion-select>
</ion-item>

<ion-item>
  <ion-label>
    <font color="#b2b2b2" size="2" align="left">
      <p align="left">* Semestre:</p>
    </font>
  </ion-label>
  <ion-select      cancelText="Fechar"      okText="Pronto"      class='largura'
disabled="{{isEnabled3}}">

```

```

<div *ngFor="let curso of courses | async">
  <ion-option *ngIf="curso.turno === providerPreferences.moreTurno &&
curso.curso === providerPreferences.moreCurso"
(ionSelect)="proc3(curso.semestre)">{{curso.semestre}}</ion-option>
</div>
</ion-select>
</ion-item>
</ion-card>

```

```

<font color="#b2b2b2" size="0,5" align="center">
  <p align="center">
    Campos com (*) são de preenchimento obrigatório.
  </p>
</font>

```

```

<div align="center">
  <button ion-button id="next" (click)="next()">
    Ver sala &nbsp; &nbsp; &nbsp;
    <ion-icon name="ios-navigate-outline"></ion-icon>
  </button>
  <br>
  <br>
  <font color="#3db7ff" size="2" (click)="corrigir()">Corrigir</font>
</div>

```

```

</ion-content>

```

cinfo.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController, ViewController }
from 'ionic-angular';

```

```
import { CursosProvider } from '../providers/cursos/cursos';
import { GlobalPreferencesVar } from '../providers/global_var/preferencias';
```

```
import { Observable } from 'rxjs/Observable';
```

```
import { CmorePage } from '../cmore/cmore';
import { CursosPage } from '../cursos/cursos';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-cinfo',
```

```
  templateUrl: 'cinfo.html',
```

```
})
```

```
export class CinfoPage {
```

```
  courses: Observable<any>;
```

```
  isEnabled1: string;
```

```
  isEnabled2: string;
```

```
  isEnabled3: string;
```

```
  constructor(public navCtrl: NavController,
```

```
    public navParams: NavParams,
```

```
    private navCtrl: ViewController,
```

```
    private toast: ToastController,
```

```
    private provider: CursosProvider,
```

```
    private providerPreferences: GlobalPreferencesVar) {
```

```
    this.courses = this.provider.getAll();
```

```
    this.isEnabled1 = 'false';
```

```
    this.isEnabled2 = 'false';
```

```
    this.isEnabled3 = 'false';
```

```
  }
```

```
  proc1(turno) {
```

```

this.providerPreferences.gravamoreTurno(turno);
this.isEnabled1 = 'true';
}

```

```

proc2(curso) {
  this.providerPreferences.gravamoreCurso(curso);
  this.isEnabled2 = 'true';
}

```

```

proc3(semestre) {
  this.providerPreferences.gravamoreSemestre(semestre);
  this.isEnabled3 = 'true';
}

```

```

corrige(){

  this.navCtrl.push(CinfoPage).then(() => {
    this.viewCtrl.dismiss();
  }) ;

  this.providerPreferences.moreTurno = '0';
  this.providerPreferences.moreCurso = '0';
  this.providerPreferences.moreSemestre = '0';

}

```

```

toHome(){

  this.viewCtrl.dismiss();

  this.providerPreferences.moreTurno = '0';
  this.providerPreferences.moreCurso = '0';
  this.providerPreferences.moreSemestre = '0';
}

```

```

next() {
  if      (this.providerPreferences.moreTurno      !=      '0'      &&
this.providerPreferences.moreCurso      !=      '0'      &&
this.providerPreferences.moreSemestre != '0') {
    this.navCtrl.push(CmorePage)
  } else {
    this.toast.create({ message: 'Por favor, preencha todos os campos acima.',
duration: 3000 }).present();
  }
}
}
}

```

cmore.html

<ion-header>

<div>

<br>

<br>

<h1>

<b>&nbsp; &nbsp; &nbsp; Resultado</b>

</h1>

<font color="#3db7ff" size="2" align="left">

<p align="left">&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; Cursos > Consulta  
Rápida > Busca por filtro</p>

</font>

</div>

</ion-header>

<ion-content padding>

<div>

<br>

<font color="#b2b2b2" size="2" align="center">

Aqui está o resultado para sua pesquisa:

&lt;/font&gt;

&lt;/div&gt;

## <ion-card>

&lt;ion-card-title&gt;

&lt;ion-item&gt;

```
<ion-icon name="ios-bookmarks" color="primary" item-start>
```

&lt;/ion-icon&gt;

```
<div *ngFor="let curso of courses | async">
```

```
<font color="#3db7ff" size="3" align="left" *ngIf="curso.turno ===  
providerPreferences.moreTurno && curso.curso === providerPreferences.moreCurso  
&& curso.semestre === providerPreferences.moreSemestre">
```

```

{{curso.sseg}}

```

&lt;/font&gt;

<font color="#b2b2b2" size="0,5" align="left">

```

{{providerPreferences.moreCurso}}

```

&lt;br&gt;{{providerPreferences.moreSemestre}}° Semestre

```
<br>{{providerPreferences.moreTurno}}
```

&lt;/font&gt;

&lt;/ion-item&gt;

&lt;/ion-card-title&gt;

&lt;/ion-card&gt;

<div align="center">

```
<button ion-button (click)="goToHome()">
```

Voltar para o início    

```
<ion-icon name="ios-school-outline"></ion-icon>
```

&lt;/button&gt;

&lt;/div&gt;

```
</ion-content>
```

```
cmore.ts
```

```
import { Component } from '@angular/core';
```

```
import { IonicPage, NavController, NavParams, ViewController } from 'ionic-angular';
```

```
import { CursosProvider } from '../providers/cursos/cursos';
```

```
import { GlobalPreferencesVar } from '../providers/global_var/preferencias';
```

```
import { Observable } from 'rxjs/Observable';
```

```
import { CursosPage } from '../cursos/cursos';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-cmore',
```

```
  templateUrl: 'cmore.html',
```

```
})
```

```
export class CmorePage {
```

```
  courses: Observable<any>;
```

```
  constructor(public navCtrl: NavController,
```

```
    public navCtrl: NavParams,
```

```
    private navCtrl: ViewController,
```

```
    private provider: CursosProvider,
```

```
    private providerPreferences: GlobalPreferencesVar) {
```

```
    this.courses = this.provider.getAll();
```

```
  }
```

```

goToHome(){

  this.providerPreferences.moreTurno = '0';
  this.providerPreferences.moreCurso = '0';
  this.providerPreferences.moreSemestre = '0';

  this.navCtrl.push('CursosPage')
  .then(=>{
    this.viewCtrl.dismiss();
  });
}

}

```

confirm.html

```

<ion-content padding>
  <ion-slides id="proc1">
    <ion-slide>
      <div class="preferences-style">
        <h1>
          <ion-icon name="ios-cafe"></ion-icon>&nbsp; &nbsp; &nbsp; Boas vindas!</h1>
        </div>
        <font color="#b2b2b2" size="2" align="left">
          <p align="center">Agora confirme as informações abaixo para poder usar nossa
interface personalizada para você!
          <br>
        </p>
        </font>
      <ion-card>
        <ion-card-title>

```



```

<ion-item>
  <ion-icon name="ios-alarm" color="primary" item-start>
  </ion-icon>
  <h2>{{providerPreferences.myTurno}}</h2>
</ion-item>
</ion-card-title>
</ion-card>

<ion-card>
  <ion-card-title>
    <ion-item>
      <ion-icon name="ios-book" color="primary" item-start>
      </ion-icon>
      <h2>{{providerPreferences.myCurso}}</h2>
    </ion-item>
  </ion-card-title>
</ion-card>

<ion-card>
  <ion-card-title>
    <ion-item>
      <ion-icon name="logo-buffer" color="primary" item-start>
      </ion-icon>
      <h2>{{providerPreferences.mySemestre}}° Semestre</h2>
    </ion-item>
  </ion-card-title>
</ion-card>

<br>

<button ion-button id="next" (click)="next()">
  Tudo certo! &nbsp; &nbsp; &nbsp;
  <ion-icon name="ios-arrow-dropright"></ion-icon>
</button>

```

```
<br><br>
```

```
<font color="#3db7ff" size="2" (click)="Corrige()">Voltar e corrigir.</font>
```

```
</ion-slide>
```

```
</ion-slides>
```

```
</ion-content>
```

```
confirm.ts
```

```
import { Component } from '@angular/core';
```

```
import { IonicPage, NavController, NavParams, ViewController, ToastController }
from 'ionic-angular';
```

```
import { GlobalPreferencesVar } from '../providers/global_var/preferencias';
```

```
import { OfflineData } from '../providers/global_var/offline_data';
```

```
import { Observable } from 'rxjs/Observable';
```

```
import { TabsPage } from '../pages/tabs/tabs';
```

```
import { SetCursoPage } from '../pages/set-curso/set-curso';
```

```
import { EventosPage } from '../eventos/eventos';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-confirm',
```

```
  templateUrl: 'confirm.html',
```

```
})
```

```
export class ConfirmPage {
```

```
  constructor(public navCtrl: NavController,
```

```
    public navCtrl: NavParams,
```

```
    private navCtrl: ViewController,
```

```
    private toast: ToastController,
```



```
</h1>
<font color="#3db7ff" size="2" align="left">
  <p align="left">&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; UnilInfo</p>
</font>
</div>
</ion-header>

<ion-content padding>

<div>
  <br>
  <font color="#b2b2b2" size="2" align="center">
    <p align="center">Aqui estão as informações do seu curso.</p>
  </font>
  <br>
</div>

<ion-card>
  <ion-card-title>
    <ion-item>
      <ion-icon name="ios-star" color="primary" item-start>
        </ion-icon>

      <div *ngFor="let curso of courses | async">
        <font color="#3db7ff" size="3" align="left" *ngIf="curso.turno ===
providerPreferences.myTurno && curso.curso === providerPreferences.myCurso &&
curso.semestre === providerPreferences.mySemestre">
          Sala: {{curso.sseg}}
        </font>
      </div>

      <font color="#b2b2b2" size="0,5" align="left">
        {{providerPreferences.myCurso}}
        <br>{{providerPreferences.mySemestre}}° Semestre
        <br>{{providerPreferences.myTurno}}
```

```

    </font>
  </ion-item>
</ion-card-title>
</ion-card>

```

```

<div>
  <font color="#b2b2b2" size="2" align="center">
    <p align="center">Ou, se preferir, consulte outro.</p>
  </font>
</div>

```

```

<ion-card (click)="selOutro()">
  <ion-card-title>
    <ion-item>
      <ion-icon name="ios-keypad-outline" color="primary" item-start>
      </ion-icon>
      <font color="#3db7ff" size="3" align="left">
        Consulta Rápida
      </font>
      <font color="#b2b2b2" size="0,5" align="left">
        <br>Pesquisar outro curso.
      </font>
    </ion-item>
  </ion-card-title>
</ion-card>

</ion-content>

```

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { GlobalPreferencesVar } from '../providers/global_var/preferencias';
import { CursosProvider } from '../providers/cursos/cursos';

import { Observable } from 'rxjs/Observable';

import { CinfoPage } from '../cinfo/cinfo';

@IonicPage()
@Component({
  selector: 'page-cursos',
  templateUrl: 'cursos.html',
})
export class CursosPage {

  courses: Observable<any>;

  constructor(public navCtrl: NavController,
    public NavParams: NavParams,
    private providerPreferences: GlobalPreferencesVar,
    private provider: CursosProvider) {

    this.courses = this.provider.getAll();
  }

  selIntro(){
    this.navCtrl.push(CinfoPage);

    this.providerPreferences.moreTurno = '0';
    this.providerPreferences.moreCurso = '0';
    this.providerPreferences.moreSemestre = '0';
  }

```

}

eventos.html

<ion-header>

## <ion-navbar>

<div>

<br>

<br>

# 

**Eventos**

# 

```
<font color="#3db7ff" size="2" align="left">
```

                UnilInfo

&lt;/font&gt;

&lt;/div&gt;

&lt;/ion-navbar&gt;

&lt;/ion-header&gt;

<ion-content padding>

<div>

<br>

<font color="#b2b2b2" size="2" align="center">

Fique por dentro do que está acontecendo pelo campus:

&lt;/font&gt;

<br>

&lt;/div&gt;

```
<ion-card *ngFor="let evento of events | async">
```

```

<ion-card-title>
  <ion-item>
    <ion-icon name="ios-quote" color="primary" item-start>
    </ion-icon>
    <font color="#3db7ff" size="3" align="left">
      {{evento.titulo}}
    </font>
    <ion-card-content style="white-space: normal;">
      <font color="#b2b2b2" size="0,5" align="right">
        <br>{{evento.descricao}}
      </font>
    </ion-card-content>
    <font color="#b2b2b2" size="0,5" align="right">
      <br>
      <b>Local:</b> {{evento.local}}
      <br>
      <b>Data:</b> {{evento.data}}
    </font>
  </ion-item>
</ion-card-title>
</ion-card>

<font color="#b2b2b2" font-size="3pt">
  <p align="right">Powered by UniInfo.</p>
</font>

</ion-content>

```

eventos.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

```



```
import { EventosProvider } from '../providers/eventos/eventos';
```

```
import { Observable } from 'rxjs/Observable';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-eventos',
```

```
  templateUrl: 'eventos.html',
```

```
})
```

```
export class EventosPage {
```

```
  events: Observable<any>;
```

```
  constructor(public navCtrl: NavController,
```

```
    public navParams: NavParams,
```

```
    private provider: EventosProvider) {
```

```
    this.events = this.provider.getAll();
```

```
  }
```

```
}
```

```
help.html
```

```
<ion-header>
```

```
<ion-navbar>
```

```
<div>
```

```
<br>
```

```
<br>
```

```
<h1>
```

```
  <b>&nbsp; &nbsp; &nbsp; Ajuda Rápida</b>
```

```
</h1>
```

```
<font color="#3db7ff" size="2" align="left">
```

```
<p align="left">&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; Suporte</p>
```

```
</font>
```

```
</div>
```

```
</ion-navbar>
```

```
</ion-header>
```

```
<ion-content padding>
```

```
<font color="#3db7ff" size="2" align="center">
```

```
<p align="center">
```

```
<b>Na configuração inicial</b>
```

```
</p>
```

```
</font>
```

```
<font color="#b2b2b2" size="2" align="left">
```

```
<p align="left">
```

```
<b>A lista está em branca:</b> Nosso aplicativo precisa de uma conexão ativa  
com a internet para que obtenha os dados
```

```
atualizados em tempo real, feche a janela de seleção verifique a conexão do  
dispositivo e abra a janela de seleção
```

```
novamente. Caso o erro persista, reinicie o aplicativo e tente novamente.
```

```
</p>
```

```
</font>
```

```
<font color="#b2b2b2" size="2" align="left">
```

```
<p align="left">
```

```
<b>Selecionei, mas não consigo avançar para a próxima tela:</b> Este erro  
ocorre devido a uma falha de gravação do dado
```

```
selecionado. Feche a janela de seleção, abra novamente, e selecione um novo  
valor. Caso o erro persista, por favor
```

```
reinicie o aplicativo e tente novamente.
```

```
</p>
```

```
</font>
```

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>Selecionei um dado errado na página anterior:</b> Ao final do processo você verá uma tela de confirmação, se estiverem

corretos clique em “Tudo certo! ”, se estiverem divergentes clique em “Voltar e corrigir. ”. Se precisar alterar seus

dados após a confirmação acesse a seção “Alterar meu perfil ” na aba “Mais ”.

</font>

<br>

<br>

<font color="#3db7ff" size="2" align="center">

<p align="center">

<b>Na aba Eventos</b>

</p>

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>A aba eventos está em branca:</b> Nosso aplicativo precisa de uma conexão ativa com a internet para que obtenha os

dados atualizados em tempo real, verifique a conexão do dispositivo e/ou reinicie o aplicativo e tente novamente.

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>Não encontrei um evento que eu vou assistir:</b> Todos os eventos são cadastrados pela equipe da secretaria do seu

campus, mas as vezes pode ser que esteja faltando algum, sendo assim entre em contato com seu coordenador para que

seja solicitado a administração de campus a inclusão do Evento.

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>Achei um evento legal, mas está com os dados errados:</b> Pode acontecer de após cadastrado um evento no sistema a

data ou alguma outra informação seja alterada. Por isso, relate ao coordenador do seu curso para que seja verificado

as informações atualizadas e solicitado a inclusão na administração do campus.

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>Perdi um evento importante pois estava com as datas ou local errado:</b> As informações são alimentadas pela secretaria

do seu campus de acordo com o calendário escolar e com os coordenadores de cursos, caso isso aconteça explique a situação

ao seu coordenador e a sua secretaria. Não nos responsabilizamos por dados inconsistentes ou desatualizados.

</font>

<br>

<br>

<font color="#3db7ff" size="2" align="center">

<p align="center">

<b>Na aba cursos</b>

</p>

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>As informações do meu curso não estão aparecendo:</b> As informações do seu curso são gravadas de acordo com o passo-a-passo

inicial do aplicativo, ou através da seção “Alterar meu perfil ” na aba “Mais ”, caso após a configuração seus dados

continuem em branco, por favor reinicie seu aplicativo e faça a configuração novamente.

</font>

<br>

<br>

<font color="#3db7ff" size="2" align="center">

<p align="center">

<b>Na aba Consulta Rápida</b>

</p>

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>A janela de seleção está em branca:</b> A ordem correta de preenchimento dos dados são: Turno > curso > semestre. Fazemos

um filtro para exibirmos apenas os dados válidos para você, por isso verifique a ordem de preenchimento e tente novamente.

Nosso aplicativo precisa de uma conexão ativa com a internet para que obtenha os dados atualizados em tempo real, feche

a janela de seleção verifique a conexão do dispositivo e abra a janela de seleção novamente. Caso o erro persista,

reinicie o aplicativo e tente novamente.

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>A janela de Seleção não exibe o turno, curso ou semestre que estou procurando:</b> Verifique se os dados foram preenchidos

corretamente, caso ele não esteja na lista verifique com a secretaria do seu campus ou com o coordenador do seu curso

e solicite a inclusão das informações no banco de dados.

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>Cliquei em “Ver Sala ”, mas está pedindo para preencher todos os campos:</b> Para exibirmos seu resultado é necessário

preencher todos os campos solicitados (\*). Caso todos os campos tenham sido preenchidos e a mensagem persista, por

favor reinicie o aplicativo e tente novamente.

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>Botão “Corrigir ” não está corrigindo:</b> Este erro se deve a um erro interno de reinicialização dos dados, por favor

reinicie o aplicativo e tente novamente.

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>Após clicar em “Corrigir ” o botão de ver sala não funciona corretamente ou me traz os dados incorretos:</b> Isso se

deve a um erro interno de gravação de dados, por favor reinicie o aplicativo e tente novamente.

</font>

<br>

<br>

<font color="#3db7ff" size="2" align="center">

<p align="center">

<b>Na aba de Resultado</b>

</p>

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>As informações não estão sendo exibidas:</b> Volte a página inicial e verifique os dados incluídos, caso os dados estejam

corretos, verifique com a secretaria do seu campus ou com o coordenador do seu curso e solicite a inclusão das informações

no banco de dados. Nosso aplicativo precisa de uma conexão ativa com a internet para que obtenha os dados atualizados

em tempo real, volte ao início, verifique a conexão do dispositivo e faça a consulta novamente.

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>As informações estão desatualizadas:</b> Pode acontecer de após cadastrado uma sala no sistema algum remanejamento

seja feito durante o semestre. Por isso, relate ao coordenador do seu curso para que seja verificado as informações

atualizadas e solicitado a inclusão na administração do campus.

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>Após clicar em “Voltar ao Início ” o botão de ver sala não funciona corretamente ou me traz os dados incorretos:</b>

Isso se deve a um erro interno de gravação de dados, por favor reinicie o aplicativo e tente novamente.

</font>

<br>

<br>

<font color="#3db7ff" size="2" align="center">

<p align="center">

<b>Na aba Locais</b>

</p>

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>A aba locais está em branca:</b> Nosso aplicativo precisa de uma conexão ativa com a internet para que obtenha os dados

atualizados em tempo real, verifique a conexão do dispositivo e/ou reinicie o aplicativo e tente novamente.

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>As informações estão desatualizadas:</b> Pode acontecer de após cadastrado um ambiente no sistema algumas informações

podem ser alteradas. Por isso, relate ao coordenador do seu curso para que seja verificado as informações atualizadas

e solicitado a inclusão na administração do campus.

</font>

<font color="#b2b2b2" size="2" align="left">

<p align="left">

<b>Não encontrei um local que eu preciso ir:</b> Todos os ambientes são cadastrados pela equipe da secretaria do seu campus,

mas as vezes pode ser que esteja faltando algum, sendo assim entre em contato com seu coordenador para que seja solicitado

a administração de campus a inclusão do ambiente.

</font>

<br>

<br>



```
<font color="#3db7ff" size="2" align="center">
```

```
<p align="center">
```

```
<b>Na aba Mais</b>
```

```
</p>
```

```
</font>
```

```
<font color="#b2b2b2" size="2" align="left">
```

```
<p align="left">
```

```
<b>Função de alterar minhas informações não funciona ou retorna um comando
inválido:</b> Este erro é causado por alguma
```

```
falha na sincronização interna do aplicativo. Por favor, reinicie o aplicativo e tente
novamente.
```

```
</font>
```

```
<font color="#b2b2b2" size="2" align="left">
```

```
<p align="left">
```

```
<b>As informações estão desatualizadas:</b> Entre em contato com seu
coordenador ou com a secretaria de seu campus e solicite
a atualização dos dados.
```

```
</font>
```

```
</ion-content>
```

```
help.ts
```

```
import { Component } from '@angular/core';
```

```
import { IonicPage, NavController, NavParams } from 'ionic-angular';
```

```
@IonicPage()
```

```
@Component({
```

```
selector: 'page-help',
```

```
templateUrl: 'help.html',
```

```

}))
export class HelpPage {

  constructor(public navCtrl: NavController, public navParams: NavParams) {
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad HelpPage');
  }

}

```

info.html

```

<ion-header>
  <ion-navbar>
    <div>
      <br>
      
      <font color="#3db7ff" size="2" align="left">
        <p align="center">Sistema de informações da Universidade Paulista Unip</p>
      </font>
    </div>
  </ion-navbar>
</ion-header>

<ion-content padding>

  <br>
  <br>

```

```

<ion-card (click)="alterar()">
  <ion-card-title>
    <ion-item>
      <ion-icon name="ios-construct" color="primary" item-start>
      </ion-icon>
      <font color="#3db7ff" size="3" align="left">
        Alterar meu perfil
      </font>
      <font color="#b2b2b2" size="0,5" align="right">
        <br>Turno | Curso | Semesetre
      </font>
    </ion-item>
  </ion-card-title>
</ion-card>
<br>

```

```

<font color="#3db7ff" size="3" align="center">
  <p align="center">Não Procure. Ache.</p>
</font>

```

```

<font color="#b2b2b2" size="2" align="right">
  <p>Destinado tanto a calouros que desejam ter acesso a localização de pontos importantes do campus como para veteranos que precisam se atentar a tudo que está ocorrendo em novidades e prazos, o UnilInfo foi desenvolvido com a mais alta segurança em dados e com o layout mais fluente, otimizado e pensado anatomicamente. Para você.</p>
</font>

```

```

<br>
<br>

```

```

<font color="#3db7ff" size="3" align="center">
  <p align="center">Precisa de ajuda?</p>

```

</font>

<ion-card (click)="exibe()">

<ion-card-title>

<ion-item>

<ion-icon name="ios-help-circle" color="primary" item-start>

</ion-icon>

<font color="#3db7ff" size="3" align="left">

Ajuda Rápida

</font>

<font color="#b2b2b2" size="0,5" align="left">

<br>Perguntas frequentes

</font>

</ion-item>

</ion-card-title>

</ion-card>

<br>

<font color="#b2b2b2" size="2" align="center">

<p>Não achou o que procura? Fale com a equipe:</p>

</font>

<font color="#3db7ff" size="2" align="center">

<p align="center">

<b>Suporte técnico</b>

</p>

</font>

<font color="#b2b2b2" size="2" align="center">

<p align="center">suporte@br.uniinfo.com</p>

</font>

<font color="#3db7ff" size="2" align="center">

<p align="center">

<b>Outros assuntos</b>

</p>

```

</font>
<font color="#b2b2b2" size="2" align="center">
  <p align="center">sac@br.uniinfo.com</p>
</font>

<br>
<br>

<ion-card>
  <ion-card-title>
    <ion-item>
      <ion-icon name="ios-people" color="primary" item-start>
      </ion-icon>
      <font color="#3db7ff" size="3" align="left">
        Nossa equipe
      </font>
      <ion-card-content style="white-space: normal;">
        <font color="#b2b2b2" size="1" align="right">
          <br>
          <b>Kaique C. Ribeiro:</b>
          <br> Desenvolvedor
          <br>
          <br>
          <b>Jorge Willian:</b>
          <br> Documentação e Planejamento
          <br>
          <br>
          <b>Leandro da Silva:</b>
          <br> Desenvolvedor web (Site promocional)
          <br>
          <br>
          <b>Wesley Aparecido:</b>
          <br> Revisor e campanhas publicitárias
        </font>

```

```

    </ion-card-content>
  </ion-item>
</ion-card-title>
</ion-card>

```

```

<br>
<br>

```

```

<ion-card>
  <ion-card-title>
    <ion-item>
      <ion-icon name="ios-code" color="primary" item-start>
    </ion-icon>
    <font color="#3db7ff" size="3" align="left">
      Sobre
    </font>
    <font color="#b2b2b2" size="0,5" align="right">
      <br>
      <b>Versão:</b> 3.5
      <br>
      <b>Atualizado em:</b> 19/05/2018
    </font>
    </ion-item>
  </ion-card-title>
</ion-card>

```

```

<br>

```

```

<font color="#b2b2b2" size="0,5" align="center">
  <p align="center">UnilInfo 2018 &#174; - Todos os direitos reservados.</p>
  <br>
</font>

```

```

</ion-content>

```

```

info.ts
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ViewController, App } from 'ionic-
angular';

import { GlobalPreferencesVar } from '../providers/global_var/preferencias';

import { HelpPage } from '../help/help';
import { SetCursoPage } from '../set-curso/set-curso';

@IonicPage()
@Component({
  selector: 'page-info',
  templateUrl: 'info.html',
})
export class InfoPage {

  constructor(public navCtrl: NavController,
    public NavParams: NavParams,
    private navCtrl: ViewController,
    public app: App,
    private providerPreferences: GlobalPreferencesVar) {

  }

  exhibe() {
    this.navCtrl.push(HelpPage);
  }

  alterar(){
    this.providerPreferences.myTurno = '0';
    this.providerPreferences.myCurso = '0';
  }
}

```

```
this.providerPreferences.mySemestre = '0';
```

```
this.app.getRootNav().setRoot(SetCursoPage);
```

```
}
```

```
}
```

linfo.html

```
<ion-header>
```

```
<ion-navbar>
```

```
<div>
```

```
<br>
```

```
<br>
```

```
<h1>
```

```
<b>&nbsp; &nbsp; &nbsp; &nbsp; {{providerLocal.localInfoAmbiente}}</b>
```

```
</h1>
```

```
<font color="#3db7ff" size="2" align="left">
```

```
<p align="left">&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; Locais</p>
```

```
</font>
```

```
</div>
```

```
</ion-navbar>
```

```
</ion-header>
```

```
<ion-content padding>
```

```
<div class="align_center">
```

```
<h1>{{providerLocal.localInfoLocalizacao}}</h1>
```

```
</div>
```

```
<font color="#4c4c4c" size="2" align="left">
```

```
<p align="center">{{providerLocal.localInfoDescricao}}</p>
```

```
<br>
```



```

    <p align="left"><b>Horário: &nbsp;  </b>{{providerLocal.localInfoHorario}}</p>
  </font>
</ion-content>

```

linfo.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

```

```

import { LocaisProvider } from '../providers/locais/locais';
import { GlobalLocalVar } from '../providers/global_var/local';

```

```

import { Observable } from 'rxjs/Observable';

```

```

@IonicPage()
@Component({
  selector: 'page-linfo',
  templateUrl: 'linfo.html',
})
export class LinfoPage {

```

```

  places: Observable<any>;

```

```

  constructor(public navCtrl: NavController,
    public NavParams: NavParams,
    private provider: LocaisProvider,
    private providerLocal: GlobalLocalVar ) {

```

```

  }

```

```

}

```

locais.html

```

<ion-header>

```

```
<ion-navbar>
  <div>
    <br>
    <br>
    <h1>
      <b>&nbsp; &nbsp; &nbsp; Locais</b>
    </h1>
    <font color="#3db7ff" size="2" align="left">
      <p align="left">&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; UnilInfo</p>
    </font>
  </div>
</ion-navbar>
</ion-header>
```

<ion-content padding>

```
<div>
  <br>
  <font color="#b2b2b2" size="2" align="center">
    <p align="center">Toque para mais detalhes:</p>
  </font>
  <br>
</div>
```

```
<ion-card *ngFor="let local of places | async" (click)="verLocal(local.ambiente,
local.localizacao, local.descricao, local.horario)">
  <ion-card-title>
    <ion-item>
      <ion-icon name="ios-compass" color="primary" item-start>
    </ion-icon>
    <font color="#3db7ff" size="3" align="left">
      {{local.ambiente}}
    </font>
    <font color="#b2b2b2" size="0,5" align="right">
```

```

        <br><b>Bloco:</b> {{local.localizacao}}
        <br><b>Horário:</b> {{local.horario}}
    </font>
</ion-item>
</ion-card-title>
</ion-card>

</ion-content>

```

locais.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { LocaisProvider } from '../providers/locais/locais';
import { GlobalLocalVar } from '../providers/global_var/local';

import { LinfoPage } from '../linfo/linfo';

import { Observable } from 'rxjs/Observable';

@IonicPage()
@Component({
  selector: 'page-locais',
  templateUrl: 'locais.html',
})
export class LocaisPage {

  places: Observable<any>;

  constructor(public navCtrl: NavController,
    public NavParams: NavParams,
    private provider: LocaisProvider,
    private providerLocal: GlobalLocalVar) {

    this.places = this.provider.getAll();
  }
}

```

```
}
```

```
verLocal(ambiente, localizacao, descricao, horario) {
  this.providerLocal.exibeAmbiente(ambiente)
  this.providerLocal.exibeLocalizacao(localizacao)
  this.providerLocal.exibeDescricao(descricao)
  this.providerLocal.exibeHorario(horario);
  this.navCtrl.push('LinfoPage');
}
```

```
}
```

set-curso.html

```
<ion-content padding>
```

```
<ion-slides id="proc1">
```

```
<ion-slide>
```

```
<div class="preferences-style">
```

```
<h1>
```

```
<ion-icon name="ios-alarm"></ion-icon>&nbsp; &nbsp; Meu turno</h1>
```

```
</div>
```

```
<font color="#b2b2b2" size="2" align="left">
```

```
<p align="center">Para personalizarmos seu ambiente precisamos que você
```

selecione o

```
<b>turno</b> do seu curso:</p>
```

```
</font>
```

```
<br>
```

```
<ion-card (click)="defineTurno('Matiturno')">
```

```
<button id="Matiturno" ion-button clear clas="width_input">Manhã</button>
```

```
</ion-card>
```

```

<ion-card (click)="defineTurno('Noturno')">
  <button id="Noturno" ion-button clear clas="width_input">Noite</button>
</ion-card>

<br><br>
  <font color="#3db7ff" size="2" (click)="ajuda()">Ajuda</font>

</ion-slide>
</ion-slides>

</ion-content>

```

set-curso.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { GlobalPreferencesVar } from '../providers/global_var/preferencias';

import { HelpPage } from '../help/help';
import { SetCurso2Page } from '../set-curso2/set-curso2';

@IonicPage()
@Component({
  selector: 'page-set-curso',
  templateUrl: 'set-curso.html',
})
export class SetCursoPage {

  constructor(public navCtrl: NavController,
    public NavParams: NavParams,
    private providerPreferences: GlobalPreferencesVar) {

```

```
}
```

```
defineTurno(val){
    this.providerPreferences.gravaTurno(val);
    this.navCtrl.push('SetCurso2Page');
}
```

```
ajuda(){
    this.navCtrl.push(HelpPage);
}
```

```
}
```

set-curso2.html

```
<ion-content padding>
  <ion-slides id="proc2">
    <ion-slide>
      <div class="preferences-style">
        <h1>
          <ion-icon name="ios-school"></ion-icon>&nbsp; &nbsp; Meu curso</h1>
        </div>
        <font color="#b2b2b2" size="2" align="left">
          <p align="center">Agora vamos precisar do seu
            <b>curso</b>:
            <br>
            <br>
          </p>
        </font>

        <ion-item>
```

```

<ion-label>
  <font color="#b2b2b2" size="2" align="left">
    <p align="left">Estudo:</p>
  </font>
</ion-label>
<ion-select cancelText="Fechar" okText="Pronto" class='largura'>
  <div *ngFor="let curso of courses | async">
    <ion-option *ngIf="curso.turno === providerPreferences.myTurno &&
curso.semestre === '1'" (ionSelect)="ready1(curso.curso)">{{curso.curso}}</ion-
option>
  </div>
</ion-select>

</ion-item>
<br>
<button ion-button id="next" (click)="next()">
  Próximo &nbsp; &nbsp; &nbsp;
  <ion-icon name="ios-arrow-dropright"></ion-icon>
</button>
<br>
<br>
<font color="#3db7ff" size="2" (click)="ajuda()">Ajuda</font>
<br>
<br>
<font color="#b2b2b2" size="1" align="left">
  <p>
    <b>*Caso seu curso não esteja na lista, entre em contato com a secretaria de
seu Campus.</b>
  </p>
</font>
</ion-slide>
</ion-slides>
</ion-content>

```

set-curso2.ts

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';
```

```
import { CursosProvider } from '../providers/cursos/cursos';
import { GlobalPreferencesVar } from '../providers/global_var/preferencias';
```

```
import { Observable } from 'rxjs/Observable';
```

```
import { SetCurso3Page } from '../set-curso3/set-curso3';
import { HelpPage } from '../help/help';
```

```
@IonicPage()
```

```
@Component({
  selector: 'page-set-curso2',
  templateUrl: 'set-curso2.html',
})
```

```
export class SetCurso2Page {
```

```
  courses: Observable<any>;
```

```
  constructor(public navCtrl: NavController,
    public navCtrl: NavParams,
    private provider: CursosProvider,
    private providerPreferences: GlobalPreferencesVar,
    private toast: ToastController) {
```

```
    this.courses = this.provider.getAll();
```

```
  }
```

```
  ajuda(){
    this.navCtrl.push(HelpPage);
```



```

    }

    goBack() {
        this.navCtrl.push('SetCursoPage');
    }

    ready1(curso) {
        this.providerPreferences.gravaCurso(curso);
    }

    next() {
        if (this.providerPreferences.myCurso != '0') {
            this.navCtrl.push('SetCurso3Page')
        } else {
            this.toast.create({ message: 'Por favor, selecione um curso.', duration: 3000
        }).present();
        }
    }
}

```

set-curso3.html

<ion-content padding>

<ion-slides id="proc1">

<ion-slide>

<div class="preferences-style">

<h1>

<ion-icon name="logo-buffer"></ion-icon>&nbsp; &nbsp; Meu semestre</h1>

</div>

<font color="#b2b2b2" size="2" align="left">

<p align="center">Por fim... Seu



**<b>\*Caso seu semestre não esteja na lista, entre em contato com a secretaria de seu Campus.</b>**

**</p>**

**</font>**

**</ion-slide>**

**</ion-slides>**

**</ion-content>**

set-curso3.ts

```
import { Component } from '@angular/core';
```

```
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';
```

```
import { CursosProvider } from '../providers/cursos/cursos';
```

```
import { GlobalPreferencesVar } from '../providers/global_var/preferencias';
```

```
import { Observable } from 'rxjs/Observable';
```

```
import { ConfirmPage } from '../confirm/confirm';
```

```
import { HelpPage } from '../help/help';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-set-curso3',
```

```
  templateUrl: 'set-curso3.html',
```

```
})
```

```
export class SetCurso3Page {
```

```
  courses: Observable<any>;
```

```
  constructor(public navCtrl: NavController,
```

```

public navParams: NavParams,
private provider: CursosProvider,
private providerPreferences: GlobalPreferencesVar,
private toast: ToastController) {

    this.courses = this.provider.getAllbyS();

}

ajuda(){
    this.navCtrl.push(HelpPage);
}

ready(semester) {
    this.providerPreferences.gravaSemestre(semester);
}

next() {
    if (this.providerPreferences.mySemestre != '0') {
        this.navCtrl.push('ConfirmPage');
    }else{
        this.toast.create({ message: 'Por favor, selecione um semestre.', duration: 3000
    }).present();
    }
}

}

```

start.html

<ion-content padding>

<ion-slides id="press" pager>

```
<ion-slide>
```

```
<div class="preferences-style">
```

```
<h1>
```

```
<ion-icon name="ios-calendar"></ion-icon>&nbsp; &nbsp; &nbsp; Eventos</h1>
```

```
</div>
```

```
<font color="#b2b2b2" size="2" align="left">
```

```
<p align="center">Agora você tem um portal direto entre a secretaria e você!  
Saiba tudo que acontece pelo Campus. Datas, eventos, prazos  
e muitas outras informações importantes filtradas exclusivamente para seu  
curso.</p>
```

```
</font>
```

```
</ion-slide>
```

```
<ion-slide>
```

```
<div class="preferences-style">
```

```
<h1>
```

```
<ion-icon name="ios-school"></ion-icon>&nbsp; &nbsp; &nbsp; Cursos</h1>
```

```
</div>
```

```
<font color="#b2b2b2" size="2" align="left">
```

```
<p align="center">Chega de pegar filas ou ficar perdido no começo do semestre  
procurando em um monte de papeis sua sala! Saia da sua  
casa já sabendo exatamente pra onde vai! Calma, se tiver alguma daquelas  
mudanças chatas durante o semestre é só  
abrir seu App e em menos de 2 segundos a informação vai estar atualizada na  
sua tela! A UnilInfo também é consciente,  
evite papeis!
```

```
</p>
```

```
</font>
```

```
</ion-slide>
```

```
<ion-slide>
```

```
<div class="preferences-style">
```

```
<h1>
```

```
<ion-icon name="ios-pin"></ion-icon>&nbsp; &nbsp; Locais</h1>
```

```
</div>
```

```
<font color="#b2b2b2" size="2" align="left">
```

```
<p align="center">"Próxima aula será na nossa quadra de esportes." Sim,
temos uma! Piscinas, três auditórios muito grandes e diversas
```

```
salas! Perdido? Também te ajudamos nisso, basta acessar nossa seção
exclusiva de locais para ver tudo que seu campus
```

```
proporciona à você.
```

```
</p>
```

```
</font>
```

```
</ion-slide>
```

```
<ion-slide>
```

```

```

```
<br>
```

```
<br>
```

```
<button ion-button (click)="configurar()">
```

```
Configurar meu perfil &nbsp; &nbsp; &nbsp;
```

```
<ion-icon name="ios-arrow-dropright"></ion-icon>
```

```
</button>
```

```
</ion-slide>
```

```
</ion-slides>
```

```
</ion-content>
```

start.ts

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
```

```
import { SetCursoPage } from '../pages/set-curso/set-curso';
```

```
@IonicPage()
@Component({
  selector: 'page-start',
  templateUrl: 'start.html',
})
export class StartPage {

  constructor(public navCtrl: NavController,
    public NavParams: NavParams) {

  }

  configurar() {
    this.navCtrl.push('SetCursoPage');
  }

}
```

tabs.html

```
<ion-tabs [selectedIndex]="1">
  <ion-tab [root]="tab1Root" tabTitle="Eventos" tabIcon="ios-calendar"></ion-tab>
  <ion-tab [root]="tab2Root" tabTitle="Cursos" tabIcon="ios-school"></ion-tab>
  <ion-tab [root]="tab3Root" tabTitle="Locais" tabIcon="ios-pin"></ion-tab>
  <ion-tab [root]="tab4Root" tabTitle="Mais" tabIcon="ios-more"></ion-tab>
</ion-tabs>
```

tabs.ts

```
import { Component } from '@angular/core';

import { EventosPage } from '../eventos/eventos';
import { CursosPage } from '../cursos/cursos';
import { LocaisPage } from '../locais/locais';
import { InfoPage } from '../info/info';
```

```
@Component({
  templateUrl: 'tabs.html'
})
export class TabsPage {

  tab1Root = EventosPage;
  tab2Root = CursosPage;
  tab3Root = LocaisPage;
  tab4Root = InfoPage;

  constructor() {

  }
}
```

provider curso.ts

```
import { Injectable } from '@angular/core';
import { AngularFireDatabase, snapshotChanges } from 'angularfire2/database';

@Injectable()
export class CursosProvider {

  private PATCH = 'c_data/';
```



```
constructor(private db: AngularFireDatabase) {
```

```
}
```

```
getAll(){
```

```
    return this.db.list(this.PATCH, ref => ref.orderByChild('curso'))
```

```
    .snapshotChanges()
```

```
    .map(changes => {
```

```
        return changes.map(c => ({
```

```
            key: c.payload.key,
```

```
            ...c.payload.val()
```

```
        }));
```

```
    })
```

```
}
```

```
getAllbyS(){
```

```
    return this.db.list(this.PATCH, ref => ref.orderByChild('semestre'))
```

```
    .snapshotChanges()
```

```
    .map(changes => {
```

```
        return changes.map(c => ({
```

```
            key: c.payload.key,
```

```
            ...c.payload.val()
```

```
        }));
```

```
    })
```

```
}
```

```
getInfoC(){
```

```
    return this.db.list(this.PATCH)
```

```
    .snapshotChanges()
```

```
    .map(changes => {
```

```
        return changes.map(c => {
```

```
            return changes.map(c => ({
```

```
                key: c.payload.key,
```

```
                ...c.payload.val()
```

```

    ));
  })
}

```

```

get(key: string){
  return this.db.object(this.PATCH + key)
    .snapshotChanges()
    .map(c => {
      return {key: c.key, ...c.payload.val()};
    })
}
}

```

provider eventos.ts

```

import { Injectable } from '@angular/core';
import { AngularFireDatabase, snapshotChanges } from 'angularfire2/database';

```

```

@Injectable()
export class EventosProvider {

```

```

  private PATCH = 'e_data';

```

```

  constructor(private db: AngularFireDatabase){

```

```

  }

```

```

  getAll(){
    return this.db.list(this.PATCH, ref => ref.orderByChild('data'))
      .snapshotChanges()
      .map(changes => {
        return changes.map(ev => ({
          key: ev.payload.key,
          ...ev.payload.val()

```

```

        }));
    })
}

get(key: string){
    return this.db.object(this.PATCH + key)
        .snapshotChanges()
        .map(ev => {
            return {key: ev.key, ...ev.payload.val()};
        })
    }
}

```

provider locais.ts

```

import { Injectable } from '@angular/core';
import { AngularFireDatabase, snapshotChanges } from 'angularfire2/database';

```

```
@Injectable()
```

```
export class LocaisProvider {
```

```
    private PATCH = 'l_data';
```

```
    constructor(private db: AngularFireDatabase){
```

```
    }
```

```
    getAll(){
```

```
        return this.db.list(this.PATCH, ref => ref.orderByChild('ambiente'))
```

```
        .snapshotChanges()
```

```
        .map(changes => {
```

```
            return changes.map(l => ({
```

```
                key: l.payload.key,
```

```

        ...l.payload.val()
    }));
}
}

```

```

get(key: string){
    return this.db.object(this.PATCH + key)
    .snapshotChanges()
    .map(l => {
        return {key: l.key, ...l.payload.val()};
    })
}
}

```

provider OFFLINE curso.ts

```
import { Injectable } from '@angular/core';
```

```
@Injectable()
```

```
export class GlobalCursoVar {
    public userName: any
    public cursoInfoTurno: any;
    public cursoInfoCurso: any;
    public cursoInfoSemestre: any;

```

```

    exibeNome(val: string){
        this.userName = val;
    }

```

```

    exibeTurno(val: string){
        this.cursoInfoTurno = val;
    }

```

```
}
```

```
provider OFFLINE local.ts
```

```
import { Injectable } from '@angular/core';
```

```
@Injectable()
```

```
export class GlobalLocalVar {
```

```
  public localInfoAmbiente: any;
```

```
  public localInfoLocalizacao: any;
```

```
  public localInfoDescricao: any;
```

```
  public localInfoHorario: any;
```

```
  exibeAmbiente(val: string){
```

```
    this.localInfoAmbiente = val;
```

```
  }
```

```
  exibeLocalizacao(val: string){
```

```
    this.localInfoLocalizacao = val;
```

```
  }
```

```
  exibeDescricao(val: string){
```

```
    this.localInfoDescricao = val;
```

```
  }
```

```
  exibeHorario(val: string){
```

```
    this.localInfoHorario = val;
```

```
  }
```

```
}
```

provider OFFLINE data.ts

```
import { Injectable } from '@angular/core';
import { Storage } from '@ionic/storage';
import { DatePipe } from '@angular/common';

import { StartPage } from '../pages/start/start';
import { TabsPage } from '../pages/tabs/tabs';

import 'rxjs/add/operator/map';
import { Tabs } from 'ionic-angular';

@Injectable()
export class OfflineData {

  //Status: string = 'StartPage';
  Status: any = StartPage;

  constructor(private storage: Storage,
    private datepipe: DatePipe) {

  }

  public changeVar() {
    this.Status = TabsPage;
  }

}
```

provider OFFLINE preferencias.ts

```

import { Injectable } from '@angular/core';
import { Storage } from '@ionic/storage';
import { DatePipe } from '@angular/common';
import 'rxjs/add/operator/map';

@Injectable()

export class GlobalPreferencesVar {
  // VARIÁVEIS DE SELEÇÃO PERMANENTES
  public myTurno: any = '0';
  public myCurso: any = '0';
  public mySemestre: any = '0';

  // VARIÁVEIS DE SELEÇÃO TEMPORÁRIA
  public moreTurno: any = '0';
  public moreCurso: any = '0';
  public moreSemestre: any = '0';

  constructor(private storage: Storage,
    private datepipe: DatePipe) {
  }

  // FUNÇÕES DE GRAVAÇÃO DE VARIÁVEIS PERMANENTES

  gravaTurno(val: string){
    this.myTurno = val;
  }

  gravaCurso(val: string){
    this.myCurso = val;
  }

  gravaSemestre(val: string){
    this.mySemestre = val;
  }

```

```
}
```

```
// FUNÇÕES DE VARIÁVEIS TEMPORÁRIAS
```

```
gravamoreTurno(val: string){  
    this.moreTurno = val;  
}
```

```
gravamoreCurso(val: string){  
    this.moreCurso = val;  
}
```

```
gravamoreSemestre(val: string){  
    this.moreSemestre = val;  
}  
}
```