

# Aplicativo mobile para avaliação de eventos e pesquisas baseado em técnicas de aprendizado de máquina

*Mobile app for event evaluation and research based on machine learning techniques*

**Alesanco Araújo Marins**

Graduando do curso de Sistemas de Informação (UNIPAM).

E-mail: alesancoam@unipam.edu.br

**José dos Reis Mota**

Professor orientador (UNIPAM).

E-mail: josereis@unipam.edu.br

---

**Resumo:** A avaliação, em todas as suas formas e meios, é muito importante. Ela se faz presente em todas as áreas, como eventos, educação, empresas, de diversas formas, como *feedback* por conversa, formulários e aplicativos mobile. Nesse contexto, este artigo relata o processo de pesquisa e desenvolvimento de um aplicativo que oferece aos envolvidos no processo avaliativo uma forma ágil e rápida de apuração de todo o processo, independentemente da forma ou meio, e, por consequência, auxilia o processo de tomada de decisão. Para o desenvolvimento do projeto, foram utilizadas tecnologias como *Ionic Framework*, linguagens de programação *Typescript* e *Python*, arquitetura *REST*, banco de dados *MYSQL*.

**Palavras-chave:** Avaliação. Aprendizado de máquina. Inteligência artificial.

**Abstract:** Evaluation, in all its forms and means, is very important. Evaluation is present in all areas, such as events, education, companies, in various forms, such as feedback by conversation, forms and mobile applications. In this context, this article reports the research and development process of an application which offers, to the ones involved in the evaluative process, an agile and fast way to verify the whole process by whatever form or medium and, consequently, help decision making process. For the project development, technologies such as *Ionic Framework*, *Typescript* and *Python* programming languages, *REST* architecture, *MYSQL* databases were used.

**Keywords:** Evaluation. Learning machine. Artificial intelligence.

---

## 1 INTRODUÇÃO

Todas as tecnologias estão em constante evolução, e isso possibilita que tudo ao seu entorno também esteja nessa constante busca por aprimoramento. O processo avaliativo de eventos e pesquisas não pode ser diferente, já que são atividades de extrema importância que possibilitam grandes experiências.

Atualmente, existem diversos meios que possibilitam avaliar um evento ou pesquisa. Os mais difundidos são por meio de formulários impressos, formulários online, redes sociais, e-mails. Contudo, não podem ser os meios mais efetivos e práticos, visto que demandam uma grande quantidade de tempo, acarretam custos extras, necessitam de uma compreensão de quem está avaliando e, por fim, existe um grande problema: os dados resultantes de todo o processo de avaliação devem ser analisados e entendidos para fornecer importantes informações ao organizador.

O uso de sistemas de apoio aprimorando os métodos promovendo maior desenvolvimento e acesso à informação. Com isso, investir em soluções inteligentes é o meio de fazer com que a avaliação de um evento ou pesquisa possa ser uma opção mais funcional.

Buscando efetivar os meios e o processo de avaliação de um evento e pesquisa, este estudo teve como objetivo geral desenvolver um aplicativo que proporcione um ambiente adequado e, por meio de perguntas e respostas, possibilite a avaliação de eventos e pesquisas, assim como a utilização de técnicas de aprendizado de máquina para classificação dos dados obtidos. Essa utilização possibilita que, por meio de um conjunto de rótulos obtidos de dados de treinamento, seja possível prever a avaliação de todos os dados obtidos, evitando diminuição de tempo gasto e minimizando riscos na tomada de decisão.

## 2 REFERENCIAL TEÓRICO

Nesta seção, são apresentados conceitos referentes ao Aprendizado de Máquina e demais tecnologias utilizadas durante o desenvolvimento do trabalho.

### 2.1 APRENDIZADO DE MÁQUINA

De acordo com Monard e Baranauskas (2003, *apud* VASSALLI, 2018), o Aprendizado de Máquina (AM) é uma área da inteligência artificial que tem como objetivo desenvolver técnicas computacionais sobre o aprendizado, além de possibilitar a criação de sistemas capazes de adquirir conhecimento de maneira automática. Cada sistema de aprendizado possui suas próprias características, não havendo, portanto, um algoritmo universal capaz de solucionar todos os problemas.

Algoritmos de AM têm sido amplamente utilizados em diversas tarefas. Essas tarefas podem ser organizadas de acordo com diferentes critérios. Um deles diz respeito ao paradigma de aprendizado a ser adotado para lidar com a tarefa. De acordo com esse critério, as tarefas de aprendizado podem ser divididas em preditivas e descritivas. Em tarefas de previsão, a meta é encontrar uma função (também chamada de modelo ou hipótese) a partir dos dados de treinamento que possa ser utilizada para prever um rótulo ou valor que caracterize um novo exemplo, com base nos valores de seus atributos de entrada e saída. Para isso, cada objeto do conjunto de treinamento deve possuir atributos de entrada e saída. Em tarefas de descrição, a meta é explorar ou descrever um conjunto de dados. Os algoritmos de AM utilizados nessas tarefas não fazem uso do atributo de saída.

A Figura 1 apresenta uma hierarquia de aprendizado de acordo com os tipos de tarefas de aprendizado.

**Figura 1** – Hierarquia de aprendizado



Fonte: Carvalho *et al.*, 2011.

No topo da Figura 1, aparece o aprendizado indutivo, processo pelo qual são realizadas as generalizações a partir dos dados. Têm-se em seguida os tipos de aprendizado supervisionado (preditivo) e não supervisionado (descritivo). As tarefas supervisionadas se distinguem pelo tipo de rótulos dos dados: discreto, no caso de classificação; e contínuo, no caso de regressão. As tarefas descritivas são genericamente divididas em agrupamento, em que os dados são agrupados de acordo com sua similaridade; sumarização, cujo objetivo é encontrar uma descrição simples e compacta para um conjunto de dados; e associação, que consiste em encontrar padrões frequentes de associações entre os atributos de um conjunto de dados.

No trabalho docente, a avaliação é uma tarefa necessária para que o avaliador não limite sua opinião a uma simples nota, e sim para que ele possa discorrer sobre o projeto avaliado em questão e para que sua avaliação seja feita de forma automática a partir disso. O foco será nas tarefas de aprendizado indutivo supervisionado, ou seja, a partir de dados já existentes. Esperam-se obter modelos preditivos em relação ao aproveitamento acadêmico dos discentes.

## 2.2 IONIC FRAMEWORK

Com a criação dos dispositivos inteligentes, surgiu uma grande quantidade de Sistemas Operacionais. Com isso, surgia também a dificuldade de padronizar uma linha de desenvolvimento de aplicações que fossem portáteis ou multiplataforma.

O ionic é um *framework* para desenvolvimento de aplicativos para dispositivos móveis que busca facilitar o desenvolvimento de aplicativos híbridos, ou seja, que funcionem independentemente de plataformas.

Segundo Gonçalves (2017), os aplicativos híbridos são desenvolvidos com os recursos de um sistema web, que podem ser empacotados e distribuídos nas lojas de aplicativos como se fossem nativos. Todo o código produzido em uma aplicação híbrida é executado dentro de um recurso chamado *webview*, um tipo especial de browser que é executado quando a aplicação híbrida é requisitada pelo usuário. Essa

forma de execução sobre o *webview* não fica explícita para o usuário, deixando a impressão de que o aplicativo se comporta da mesma forma que um aplicativo nativo. O nome híbrido é dado justamente pela característica da junção de código nativo para empacotamento e distribuição do aplicativo com o código não nativo (HTML, CSS e JavaScript), responsável pelo visual e funcionalidades da aplicação.

## 2.3 REDES NEURAIAS

De acordo com Haykin (2007), uma rede neural é um processador maciça e paralelamente distribuído e constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso.

As redes neurais se assemelham bastante ao funcionamento do cérebro humano. Assim, é possível traçar um paralelo entre ambos, da seguinte forma: no momento de seu nascimento, um cérebro tem uma grande estrutura e habilidade para desenvolver suas próprias regras através do que usualmente denominamos "experiência". Na verdade, a experiência vai sendo acumulada com o tempo, sendo que o mais dramático desenvolvimento (i.e., por ligações físicas) do cérebro humano acontece durante os dois primeiros anos de vida, mas o desenvolvimento continua para muito além desse estágio.

Um neurônio em "desenvolvimento" é sinônimo de um cérebro plástico: a plasticidade permite que o sistema nervoso em desenvolvimento se adapte ao seu meio ambiente. Assim como a plasticidade parece ser essencial para o funcionamento dos neurônios como nas unidades de processamento de informação do cérebro humano, também ela o é com relação às redes neurais construídas com neurônios artificiais (HAYKIN, 2007).

### 2.3.1 REDE NEURAL DO TIPO LSTM

Uma Rede de Memória Longa de Curto Prazo, mais conhecida como Long Short-Term Memory (LSTM), é uma Rede Neural Recorrente (RNN, de Recurrent Neural Network) composta por unidades LSTM.

A ideia principal de uma Rede Neural Recorrente é que ela pode ser alimentada com dados importantes ao modelo – no caso, os dados históricos de avaliações realizadas pelo Centro Universitário de Patos de Minas (UNIPAM), para realizar as previsões. É possível definir que o modelo receberá entradas de todo o histórico de avaliações e retornará uma nota para as seguintes. Contudo, sem o treinamento, o modelo não é capaz que realizar a previsão. É necessário treinar o modelo com um histórico de avaliações por diversas vezes, para que ele seja calibrado e, assim, espere-se que, no fim, preveja uma nota para determinada entrada.

Elman (1990) formalizou uma RNN da seguinte forma:

$$h^t = \sigma(W_h X + W_{hh} h^{t-1})$$

$$y = \sigma(W_y h^t)$$

em que  $h$  é definido como o estado oculto,  $\sigma$  é a função de ativação,  $W$  são os pesos,  $X$  entrada e  $y$  saída.

Já uma rede neural do tipo LSTM, segundo Vassalli (2018), é composta de um neurônio e portões de entrada, esquecimento e saída. O neurônio é responsável pela memória da rede, representada pela ativação baseada na soma ponderada, e os portões são uma forma de, opcionalmente, permitir a informação. Vamos analisar uma rede LSTM em seus componentes:

- estado oculto: usado para determinar o que esquecer, entrar e sair no passo seguinte,
- estado de entrada: combinação do estado oculto e entrada atual,
- estado interno: valores com função de memória,
- portão de entrada: decide se a entrada alcança o estado interno

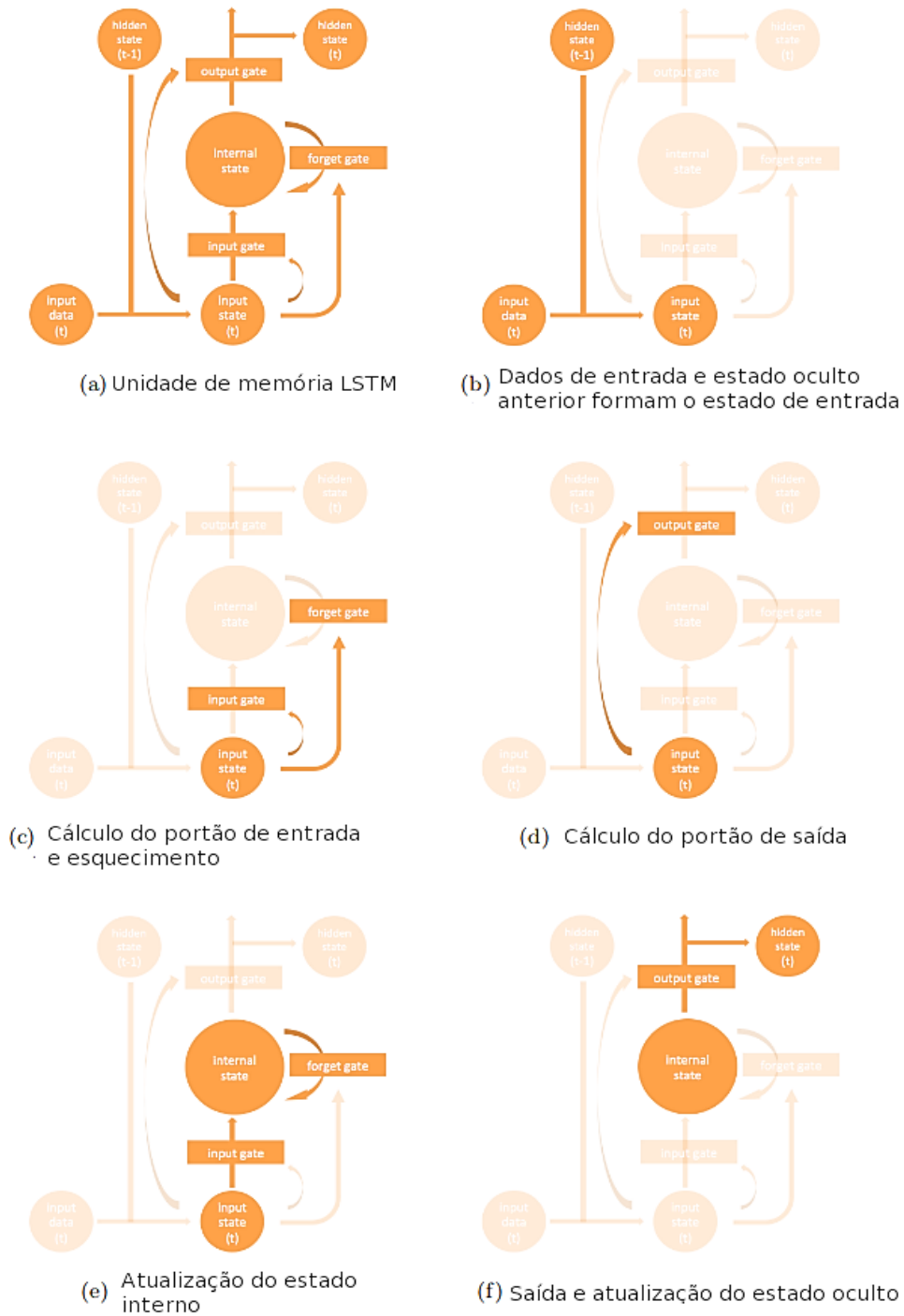
$$i^{(t)} = \sigma(W^{ix}x^{(t)} + W^{ih}h^{(t-1)} + b_i)$$

- portão de esquecimento: decide se o estado interno desconsidera o anterior
- portão de saída: decide se o estado interno é passado à saída e ao estado oculto no passo seguinte.

$$o^{(t)} = \sigma(W^{ox}x^{(t)} + W^{oh}h^{(t-1)} + b_o)$$

A Figura 2 mostra o esquema de uma unidade LSTM e suas funções detalhadas.

**Figura 2 – A unidade LSTM e suas funções detalhadas**



Fonte: Wang; Raj, 2017

O funcionamento é baseado na conexão entre os neurônios de RNN, que devem ser conectados entre as camadas, podendo ser anteriores e seguintes. Para cada neurônio a partir de entradas recebidas, é atribuído um determinado peso e subtraído deles um valor definido como bias e, por fim, submetidos à função de ativação para representar a ativação do neurônio. Os pesos constituem a memória da rede a partir do treinamento recebido.

### 3 METODOLOGIA

Inicialmente, foi realizado um estudo bibliográfico sobre a Inteligência Artificial, aprofundando na área do Aprendizado de Máquina, buscando esclarecer conceitos sobre os algoritmos que foram utilizados na análise dos dados. Também foi feito um estudo sobre as demais tecnologias a serem utilizadas no desenvolvimento. Na segunda etapa, foi feito um levantamento dos requisitos funcionais e não funcionais, buscando elucidar e definir as necessidades do negócio, para o qual o aplicativo deve prover a solução. Na terceira etapa, foi realizada a modelagem do modelo do banco de dados, buscando criar um modelo de acordo com os requisitos definidos na etapa anterior. Na quarta etapa, foi desenvolvido o aplicativo juntamente com a API, buscando implementar e testar as principais operações CRUD definidas no levantamento de requisitos. Na última etapa, foi desenvolvido o modelo de aprendizado de máquina juntamente com sua API e, após o desenvolvimento, foram realizados testes e ajustes buscando adequá-lo aos dados gerados pelo aplicativo.

O Quadro 1 lista ferramentas e tecnologias utilizadas no desenvolvimento do aplicativo.

**Quadro 1** – Ferramentas e tecnologias utilizadas

Nome	Descrição
Visual Studio Code	Editor de código utilizado para criar e depurar aplicativos da Web e da nuvem.
SQL Power Architect	Software utilizado para desenvolver os diagramas UML.
Ionic	Framework para desenvolvimento de aplicações voltadas para dispositivos móveis.
Mysql	Sistema gerenciador de banco de dados que utiliza a linguagem SQL para gerenciar o conteúdo do banco de dados.
NodeJS	Interpretador de código JavaScript com o código aberto voltado para o lado do servidor.
Python	Linguagem de programação de alto nível, interpretada, orientada a objetos, funcional e de tipagem dinâmica e forte.
Firebase Cloud Message	Solução de mensagens entre plataformas que permite o envio confiável de notificações sem custo.
BitBucket	Serviço de hospedagem de repositório para versionamento do código.
Microsoft Office Word	Editor de texto utilizado para documentação do projeto.
Google Cloud	Plataforma de serviços em nuvem oferecida pela Google.

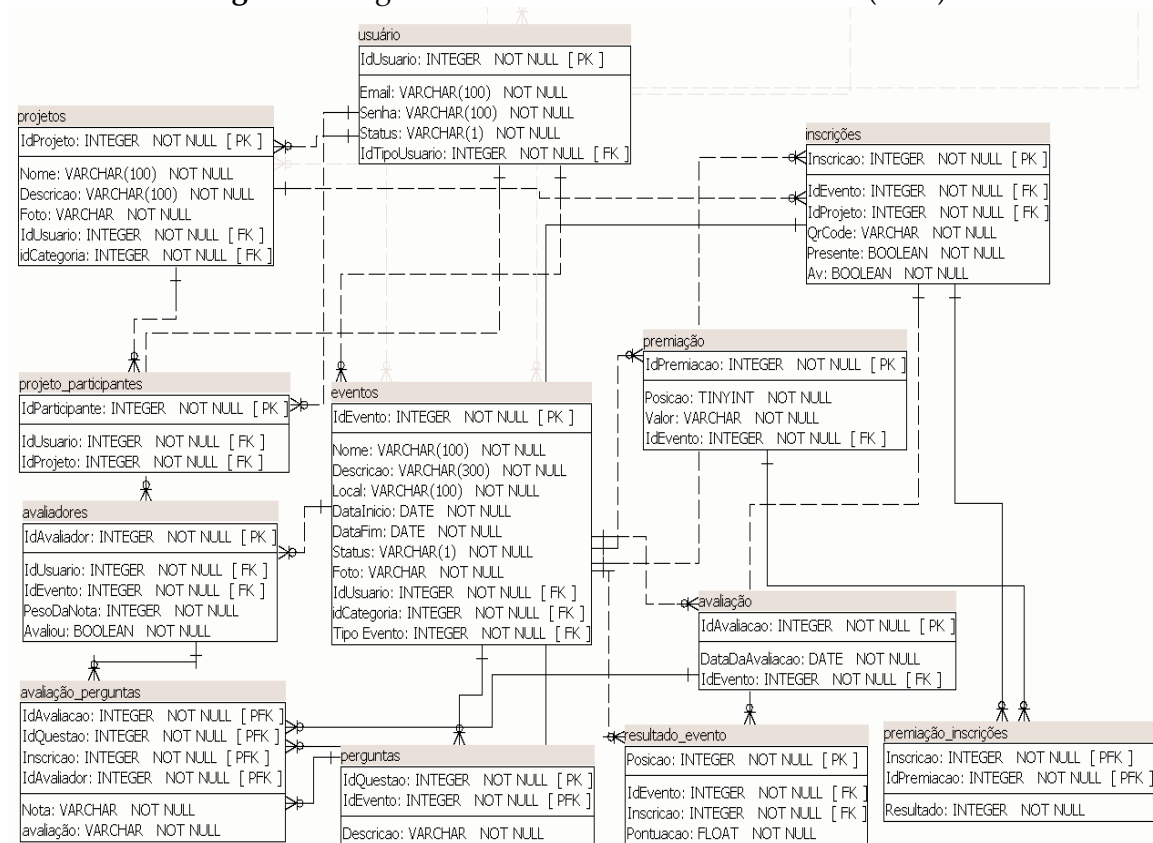
**Fonte:** Elaborado pelos autores, 2018.

#### 4 DESENVOLVIMENTO E RESULTADOS

Inicialmente, foi realizada uma pesquisa nos sistemas já existentes e no processo de avaliação utilizado no Projeto Integrador do Centro Universitário de Patos de Minas (UNIPAM), para definir quais os requisitos necessários para o desenvolvimento do aplicativo, tendo em vista que seu principal objetivo é automatizar a avaliação.

A partir dos requisitos levantados, foi elaborado o Diagrama de Entidade e Relacionamento (DER), para descrever os objetos (entidades) envolvidos no domínio de negócio, com suas características (atributos) e como elas se relacionam entre si (relacionamentos). A Figura 3 apresenta o Diagrama de Entidade e Relacionamento (DER) elaborado.

**Figura 3.** Diagrama de Entidade e Relacionamento (DER)



**Fonte:** Elaborado pelos autores, 2018.

Entre as entidades destacam-se as mais importantes para funcionamento do aplicativo, como a entidade usuário, responsável por manter as informações do usuário e controlar o acesso ao aplicativo logo que as principais funcionalidades estão restritas a usuários cadastrados. A entidade usuário está diretamente relacionada às entidades evento e projeto, pois, para criar novos projetos e eventos, o usuário já deve estar previamente cadastrado no aplicativo. A entidade evento é relacionada com as entidades avaliador, pergunta, inscrição, premiação, resultado e avaliação. Ao criar um evento, será necessário informar os avaliadores, com suas respectivas notas, as

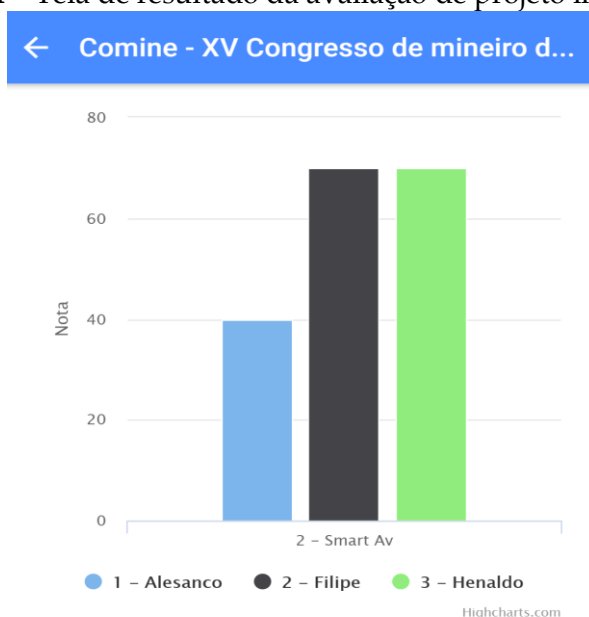


perguntas para avaliação e sua premiação por posição caso possua. A entidade projeto possibilita atribuir diversos participantes, possibilitando que a avaliação não seja feita somente individual, mas também por grupos.

Outra entidade chave para o aplicativo é a avaliação\_pergunta. A partir dos dados obtidos da avaliação de cada avaliador, é realizada uma chamada a API que realiza o machine learning em cada avaliação, gerando, assim, os resultados para a entidade resultado\_evento.

Após a apuração do evento, é possível a cada inscrito bem como seus participantes analisar os resultados da avaliação detalhada e suas respectivas notas, conforme mostra a Figura 4.

**Figura 4** – Tela de resultado da avaliação de projeto individual



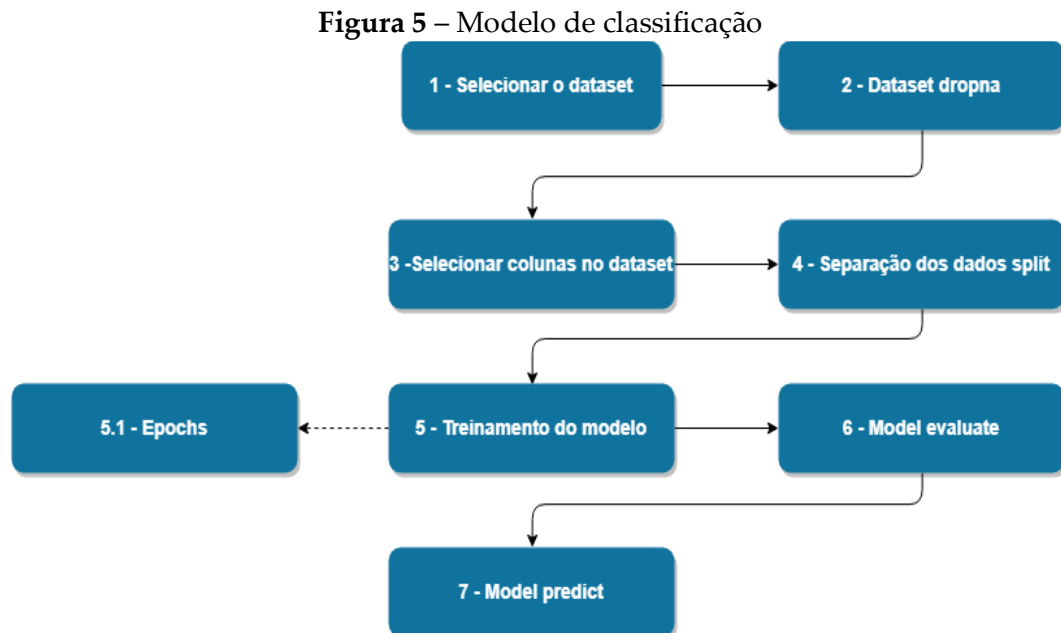
**Fonte:** Dados da pesquisa, 2018.o

Para proporcionar uma melhor comunicação entre usuário e aplicativo, foram desenvolvidas funções para envio de notificações automáticas utilizando *Firebase Cloud Messaging* (FCM). Cada função consiste em um conjunto de instruções a serem executadas quando suas condições definidas forem satisfeitas. As condições definidas variam de acordo com seu objetivo, podendo ser: a) evento iniciado; b) evento finalizado; c) projeto premiado. Quando uma das condições é satisfeita, a função responsável envia a notificação ao FMC através do método *sendToDevice()* e ele fica encarregado de enviar a notificação para o aplicativo.

Para que fosse possível a realização do *machine learning*, o Centro Universitário de Patos de Minas (UNIPAM) disponibilizou dados obtidos de vários tipos de avaliações realizadas semestralmente, por meio do portal acadêmico. Os dados foram utilizados para treinamento do algoritmo de aprendizado de máquina e validação dos resultados.

Após a obtenção e adequação dos dados, foi criado um modelo utilizando algoritmo de classificação, técnica de aprendizado supervisionado que, através de

conjunto de padrões anteriormente classificados (dados de treinamento), consegue prever uma resposta a partir de um conjunto de dados totalmente novo. A Figura 5 representa o modelo.



Fonte: Elaborado pelos autores, 2018.

O modelo da Figura 5 é descrito em etapas. Na primeira etapa, é selecionado o *dataset* com os dados obtidos. Na segunda etapa, são feitas diversas tratativas para modelar os dados de acordo com o modelo, buscando retirar dados inconsistentes a partir do método *dropna*, para obter uma predição assertiva e mais próxima da realidade. Na terceira etapa, são escolhidas as variáveis que serão trabalhadas pelo modelo; são selecionadas através do comando `data[['coluna1', 'coluna2']]` e, após essa seleção, é criado um novo *dataset*, filtrando apenas as colunas e valores das variáveis definidas. Na quarta etapa, os dados do *dataset* são divididos em dois novos *datasets*, para que, posteriormente, um deles seja dedicado apenas para o treinamento e o outro para validação dos resultados obtidos. O *dataset* foi dividido em 50% para treinamento e 50% para validação. Na quinta etapa, é realizado o treinamento do modelo; nesta etapa, foram definidos 15 *epochs*, que consistem na quantidade de vezes que o algoritmo irá passar pelos dados. Durante cada *epoch*, é possível acompanhar o aumento da acurácia do modelo conforme a Figura 6 apresenta.

**Figura 6 – Evolução do modelo**

```
In [9]: batch_size = 32
model.fit(X_train, Y_train, epochs = 15, batch_size=batch_size, verbose = 2)

Epoch 1/15
- 240s - loss: 0.4299 - acc: 0.8074
Epoch 2/15
- 238s - loss: 0.2595 - acc: 0.8957
Epoch 3/15
- 236s - loss: 0.1903 - acc: 0.9287
Epoch 4/15
- 236s - loss: 0.1515 - acc: 0.9442
Epoch 5/15
- 243s - loss: 0.1212 - acc: 0.9549
Epoch 6/15
- 251s - loss: 0.1114 - acc: 0.9587
Epoch 7/15
- 249s - loss: 0.0914 - acc: 0.9653
Epoch 8/15
- 237s - loss: 0.0833 - acc: 0.9708
Epoch 9/15
- 236s - loss: 0.0657 - acc: 0.9770
Epoch 10/15
- 234s - loss: 0.0598 - acc: 0.9781
Epoch 11/15
- 233s - loss: 0.0595 - acc: 0.9778
Epoch 12/15
- 254s - loss: 0.0529 - acc: 0.9806
Epoch 13/15
- 276s - loss: 0.0529 - acc: 0.9820
Epoch 14/15
- 278s - loss: 0.0491 - acc: 0.9850
Epoch 15/15
- 277s - loss: 0.0451 - acc: 0.9863
```

**Fonte:** Elaborado pelos autores, 2018.

Ao final dos 15 *epochs*, é possível mensurar que o modelo conseguiu uma acurácia de 0.9863 com os dados de treinamento. Na quinta e última etapa, a partir da entrada de dados, o modelo irá predizer a sua devida nota. Após a predição, sua saída é ajustada ao peso da nota de cada avaliador e alimentará o resultado do evento.

## 5 CONCLUSÃO

O estudo realizado possibilitou o desenvolvimento de um aplicativo para avaliação de eventos e pesquisas baseado em técnicas de aprendizado de máquina. O uso das técnicas de aprendizado de máquina para avaliação obteve resultado positivo, possibilitando não apenas uma agilidade no processo avaliativo, mas também uma melhora do processo de tomada de decisão de forma assertiva.

Durante o desenvolvimento, um dos obstáculos não somente para o trabalho em questão, mas também para todas as áreas relacionadas à inteligência artificial se fez presente, a obtenção de dados suficientes para treinamento e avaliação do modelo. Os dados obtidos foram disponibilizados pelo Centro Universitário de Patos de Minas (UNIPAM). Os dados foram separados em dois conjuntos, um conjunto para treinamento e outro para validação. Ao final, foi possível mensurar que a acurácia obtida pelo modelo foi satisfatória.

Futuramente, pretende-se que o aplicativo não se limite a avaliações de eventos e pesquisas, mas busque auxiliar na área acadêmica por meio da avaliação de trabalhos e testes acadêmicos.

## REFERÊNCIAS

CARVALHO, André Ponce de Leon *et al.* **Inteligência artificial**: uma abordagem de Aprendizado de Máquina. Rio de Janeiro: Editora LTC Editora, 2011.

ELMAN, J. L. **Finding structure in time**. *Cognitive Science*, 14(2), 179–211, 1990.

FACELI, Katti; LORENA, Ana Carolina; GAMA, João; CARVALHO, André Carlos Ponce de Leon Ferreira de. **Inteligência artificial**: uma abordagem de aprendizado de máquina. Rio de Janeiro: LTC, 2011.

GOIS, Adrian. **Ionic Framework**: construa aplicativos para todas as plataformas mobile. [S.l.]: Casa do Código, 2017. v. 1.

GONÇALVES, Allan Jheyson Ramos. Desenvolvimento de aplicativos híbridos com o ionic framework. **Escola regional de informática do Piauí**. livro anais – artigos e minicursos, Piauí, v. 1, n. 1, p. 500-515, jun. 2017. Disponível em: <http://www.eripi.com.br/2017/images/anais/minicursos/13.pdf>. Acesso em: 25 fev. 2018.

HAYKIN, Simon. **Redes neurais**: princípios e prática. 2. ed. [s. L.]: Bookman Editora, 2007.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. *In*: VASSALLI, Lucas Coutinho. **Aplicação de Redes Neurais LSTM para a previsão de curto prazo de vazão do Rio Paraíba do Sul**. 2018. 36 f. Tese (Doutorado) - Curso de Engenharia Computacional, Universidade Federal de Juiz de Fora, Juiz de Fora, 2018.

WANG, H.; RAJ, B. **On the origin of deep learning**. ArXiv e-prints, fev. 2017.