

**Universidade São Tomás de Moçambique**

**Faculdade de Ciências e Tecnologias de Informação**

Dércio Pedro Vicente Nacare

**Protótipo de uma Aplicação web para o controlo de assiduidade dos funcionários da USTM**

Maputo, Abril de 2020



**Universidade São Tomás de Moçambique**

**Faculdade de Ciências e Tecnologias de Informação**

Dércio Pedro Vicente Nacare

**Protótipo de uma Aplicação web para o controlo de assiduidade dos funcionários da**

**USTM**

Supervisor: Miguel Momade

Monografia para obtenção do grau de Licenciatura em Administração de Sistemas de Informação e Redes

Maputo, Abril de 2020

# **DEDICATÓRIA**

Dedico este trabalho aos meus pais, Vicente Pedro Nacare e Beatriz Eugênio Manjate.

# **AGRADECIMENTOS**

Agradeço a Deus e a minha família em especial aos meus pais e irmãos pelo apoio no crescimento escolar.

Aos meus professores, colegas e amigos que com suas competências contribuíram e muito no decorrer do desenvolvimento deste trabalho.

# **ÍNDICE DE FIGURAS**

[Figura 1 - Processamento de dados para geração de informação (Fonte: autor) 13](#_Toc52725982)

[Figura 2 - Exemplo de representação de diagrama de classe (Fonte: Autor) 17](#_Toc52725983)

[Figura 3 - Exemplo de representação de diagrama de classe (Fonte: Autor) 19](#_Toc52725984)

[Figura 4 - Exemplo de representação de Diagrama de Actividades (Fonte: Autor) 21](#_Toc52725985)

[Figura 5 - Exemplo de Diagrama de sequência de eventos(Fonte: Autor) 23](#_Toc52725986)

# **LISTA DE ABREVIATURAS**

|  |  |
| --- | --- |
| DER | Diagrama de Entidade-Relacionamento |
| IDE | *Integrated Development Environment* |
| IG | Interface Gráfica |
| RH | Recursos Humanos |
| SGBD | Sistema de Gestão de Banco de Dados |
| SI | Sistema de Informação |
| STS | *Spring Tool Suite* |
| TI | Tecnologias de Informação |
| UML | *Unified Modeling Language* |
| USTM | Universidade São Tomás de Moçambique |

**Índice**

[**DEDICATÓRIA** ii](#_Toc52725836)

[**AGRADECIMENTOS** iii](#_Toc52725837)

[**ÍNDICE DE FIGURAS** iv](#_Toc52725838)

[**LISTA DE ABREVIATURAS** v](#_Toc52725839)

[**I.** **INTRODUCÇÃO** 8](#_Toc52725840)

[**1.1.** **Contextualização** 8](#_Toc52725841)

[**1.2.** **Justificativa** 9](#_Toc52725842)

[**1.3.** **Problema** 9](#_Toc52725843)

[**1.3.1.** **Pergunta de pesquisa** 10](#_Toc52725844)

[**1.4.** **Hipóteses** 10](#_Toc52725845)

[**1.5.** **Objectivos** 11](#_Toc52725846)

[**1.5.1.** **Objectivo Geral** 11](#_Toc52725847)

[**1.5.2.** **Objectivos específicos** 11](#_Toc52725848)

[**1.6.** **Organização do trabalho** 11](#_Toc52725849)

[**II.** **REVISÃO BIBLIOGRÁFICA** 12](#_Toc52725850)

[**2.1.** **Conceitos** 12](#_Toc52725851)

[**2.1.1.** **Dados** 12](#_Toc52725852)

[**2.1.2.** **Informação** 12](#_Toc52725853)

[**2.2.** **Sistemas de informação (SI)** 12](#_Toc52725854)

[**2.2.** **Aplicação web** 13](#_Toc52725855)

[**2.3.** **Protótipo** 14](#_Toc52725856)

[**2.3.1.** **Classificação de Protótipos** 14](#_Toc52725857)

[**2.4.** ***Unified Modeling Language* (UML)** 15](#_Toc52725858)

[**2.4.1.** **Diagramas da UML** 16](#_Toc52725859)

[**III.** **METODOLOGIA** 24](#_Toc52725860)

[**3.1.** **Metodologia de Pesquisa** 24](#_Toc52725861)

[**3.1.1.** **Quanto a natureza** 24](#_Toc52725862)

[**3.1.2.** **Quanto ao procedimento técnico** 24](#_Toc52725863)

[**3.1.3.** **Quanto a abordagem** 24](#_Toc52725864)

[**3.1.4.** **População e amostra** 25](#_Toc52725865)

[**3.1.5.** **Técnica de recolha de dados** 25](#_Toc52725866)

[**3.2.** **Metodologia de Desenvolvimento** 26](#_Toc52725867)

[**3.2.1.** ***Integrated Development Environment* (IDE)** 26](#_Toc52725868)

[**3.2.2.** **Modelagem do Sistema** 26](#_Toc52725869)

[**3.2.3.** **Linguagem de programação** 26](#_Toc52725870)

[**3.2.4.** **Sistema de Gestão de Banco de dados** 27](#_Toc52725871)

[**3.2.5.** **Sistema de controlo de versão** 27](#_Toc52725872)

[**IV.** **CONCLUSÕES E RECOMENDAÇÕES** 28](#_Toc52725873)

[**4.1.** **Conclusões** 28](#_Toc52725874)

[**4.2.** **Recomendações** 28](#_Toc52725875)

[**V.** **REFERENCIAS BIBLIOGRÁFICAS** 29](#_Toc52725876)

# **INTRODUCÇÃO**

## **1.1. Contextualização**

Com o desenvolvimento da tecnologia, os Sistemas de Informação (SI) tornam-se cada vez mais presentes na vida dos colaboradores bem como das organizações, visto que “Os Sistemas de informação para além de dar apoio à tomada de decisões, à coordenação e ao controlo, esses sistemas também auxiliam os gerentes e trabalhadores a analisar problemas, visualizar assuntos complexos e criar novos produtos” (Laudon & Laudon, 2010, p. 12).

Por outro lado, as tecnologias de informação por si só não garantem o sucesso e a sobrevivência de uma empresa, pois de acordo com a pesquisa da *Global Human Trends*, realizada pela *Deloitte* em 2017, quase 80% dos líderes de Recursos Humanos (RH) consideram a experiência do trabalhador como uma tendência importante para a sobrevivência da empresa (Santos, 2017).

Devido à importância que os trabalhadores apresentam para as empresas e tendo em conta que o registo da entrada, saída e controlo de faltas é uma necessidade recorrente dos RH, torna-se imprescindível a utilização de sistemas de controlo de assiduidade para a planificação financeira e pagamento justo de salários aos trabalhadores. Porém, muitas das empresas têm usado soluções pouco modernas para este processo baseando-se no uso de livros de ponto para o registo da hora de entrada e saída, onde apresenta-se como um processo centralizado, pouco confiável, susceptível a fraudes podendo gerar custos extras devido a má conservação e trabalho extra para a administração no processamento dos dados.

Deste modo, os departamentos de RH das empresas precisam adotar formas e estratégias sofisticadas para gerir e melhorar a experiência dos seus trabalhadores, como forma de contribuir para o crescimento da empresa, criando condições para inibir processos fraudulentos evitando assim conflitos de interesses entre a entidade patronal e os trabalhadores.

Neste contexto, olhando para aquilo que é a realidade da Universidade São Tomás de Moçambique (USTM) no que se refere a submissão dos funcionários a procederem com assinatura do livro de ponto e tendo em conta a inexistência de meios sofisticados para o efeito, o presente trabalho visará a realização uma pesquisa sobre o processo de registo de entradas e saídas no livro de ponto por parte dos funcionários a tempo inteiro da USTM, em que propõe-se o desenvolvimento de um protótipo de uma Aplicação web para auxiliar neste processo.

## **1.2. Justificativa**

Tendo observado o processo de assinatura do livro de ponto na USTM, onde resultou na constatação da violação das regras pré-estabelecidas para este processo, na qual alguns funcionários tem procedido com essa actividade de forma fraudulenta em benefício próprio.

E a instituição por falta de conhecimento ou por não dispor de mecanismos que possam inibir este tipo de acção tendo como principal base o uso de soluções baseadas nas Tecnologias de Informação (TI), poderão estar a somar prejuízos em relação ao pagamento injusto de salários aos funcionários fraudulentos, pois, o processamento de dados do livro de ponto para testar a assiduidade do funcionário poderá estar a resultar em informações não verídicas, muitas das vezes beneficiando o funcionário e lesando financeiramente a instituição.

## **1.3. Problema**

A USTM, uma instituição de ensino superior, na qual dedica-se na formação e capacitação profissional de estudantes em diversas áreas de conhecimento científico e em níveis de licenciatura e mestrado, para além de dispor de funcionários a tempo parcial que compreende o grupo de docentes, dispõe de funcionários a tempo inteiro.

Onde os mesmos estão afectos a diferentes unidades orgânicas, onde exercem as suas funções em período compreendido entre as 8 horas da manhã às 16 horas, deste modo os funcionários são requeridos a procederem com a assinatura do livro de ponto para registar a sua entrada pelas 8 horas de cada dia laboral de trabalho com uma tolerância de até 30 minutos de atraso.

Este livro é disponibilizado na secretaria geral e tem havido diversos problemas em relação a disponibilidade do mesmo para a assinatura da hora de entrada por parte de cada funcionário da instituição, pois:

* Os atrasos da responsável pela secretaria geral têm refletido diretamente na disponibilidade do livro para assinatura como também no processo de colocação de nomes no livro ponto, que feito pelos Recursos Humanos (RH) no primeiro dia de semana laboral de trabalho.
* Alguns funcionários por vezes chegam a tempo e, no entanto não conseguem proceder com a assinatura da entrada devido à indisponibilidade do livro de ponto, podendo os mesmos estar sob o risco de terem a marcação de falta e pagamento injusto de salário.
* Pelas 8 horas e 30 minutos os RH deverão proceder com a retirada do livro de ponto da secretaria geral como forma de inibir que funcionários atrasados possam proceder com a assinatura da entrada, o que nem sempre ocorre conforme o estabelecido, pois em alguns casos é possível encontrar disponível o livro até às 9 horas, perfazendo assim mais de 60 minutos de tolerância de atraso.

Por outro lado, para além da assinatura da entrada por parte dos funcionários, pelas 16 horas de cada dia de trabalho, é necessário proceder com a assinatura da saída, onde também tem apresentado alguns problemas para a instituição, pois:

* Não há mecanismos de controlo do livro a essa hora, o que acaba por ocasionar em casos de funcionários que procedem com a assinatura da saída e adiantam a assinatura da entrada do dia seguinte para que possam chegar atrasados ou até mesmo faltarem e sobre eles não ocorrer a marcação de falta.
* Alguns funcionários acabam saindo mais cedo do trabalho não assinando a hora de saída e, no entanto no dia seguinte para além de procederem com a assinatura da entrada, procedem com a assinatura de saída do dia anterior para que sobre eles não ocorra a marcação de falta.

### **1.3.1. Pergunta de pesquisa**

De que forma pode-se tonar simples, confiável e rápido o processo de registo de entrada e saída dos funcionários a tempo inteiro da Universidade São Tomas de Moçambique através de uma aplicação de controlo de assiduidade?

## **1.4. Hipóteses**

**H1** – A aplicação irá permitir com que os RH possam obter dados fidedignos em relação a assiduidade dos funcionários, sem a necessidade de efectuar a recolha de dados diariamente em livros de ponto, inibindo acções fraudulentas e proporcionando o pagamento justo de salários.

**H2** – A aplicação poderá proporcionar aos funcionários um ambiente automatizado para o registo da hora de entrada e saída, reduzindo o tempo de espera e movimentação do livro de ponto entre o sector dos RH e Secretaria Geral.

## **1.5. Objectivos**

### **1.5.1. Objectivo Geral**

Desenvolver o protótipo funcional de uma Aplicação web para o controlo de assiduidade dos funcionários a tempo inteiro da USTM.

### **1.5.2. Objectivos específicos**

* Apresentar a fundamentação teórica do problema de pesquisa;
* Descrever os principais constrangimentos do processo actual no controlo de assiduidade;
* Propor um novo modelo para o processo de controlo de assiduidade;
* Desenvolver um protótipo funcional para o modelo proposto;

## **1.6. Organização do trabalho**

O presente trabalho está organizado em capítulos, nomeadamente:

**CAPITULO I: INTRODUCÇÃO**

Faz à apresentação do trabalho, contendo a contextualização, justificação, a definição do problema, a pergunta de pesquisa, hipóteses, objectivo geral e objetivos específicos que conduziram o trabalho.

**CAPITULO II: REVISÃO BIBLIOGRÁFICA**

Apresenta a fundamentação teórica do trabalho, onde encontramos os principais conceitos, classificações e pesquisas relacionadas com o tema do trabalho.

**CAPITULO III: METODOLOGIA**

Apresenta os passos seguidos para elaboração do trabalho, onde foi usada a metodologia de pesquisa, metodologia de desenvolvimento, algumas ferramentas de tratamento de dados e de desenvolvimento.

# **REVISÃO BIBLIOGRÁFICA**

Segundo (Pradanov & Freitas, 2013), “Após a escolha do tema, o pesquisador deve iniciar amplo levantamento das fontes teóricas (relatórios de pesquisa, livros, artigos científicos, monografias, dissertações e teses), com o objectivo de elaborar a contextualização da pesquisa e seu embasamento teórico.”

Deste modo o presente capítulo apresentará o co-relacionamento da pesquisa com o universo teórico, onde serão expostos alguns conceitos relacionados com a pesquisa com vista o alcance de uma melhor qualidade do mesmo.

## **2.1. Conceitos**

### **2.1.1. Dados**

De acordo com o pensamento de Laudon e Laudon (2010), Dados são sequências de fatos ainda não analisados, representativos de eventos que ocorrem nas organizações ou em ambiente físico, antes de terem sido organizados e arranjados de uma forma que as pessoas possam entendê-los e usá-los.

Isto é, dados ainda não processados não podem ser usados como veículo para a tomada de decisão, pois por si só, eles não representam significado algum.

### **2.1.2. Informação**

Denomina-se informação a um conjunto de dados que foram previamente processados/estruturados numa forma que os torna imprescindíveis ao processo de tomada de decisão, podendo ser também definida como sendo a entidade que mede a diferença entre o saber e não saber, entre conhecer e não conhecer. (Caldeira, 2011)

## **2.2. Sistemas de informação (SI)**

Um Sistema de Informação pode ser definido tecnicamente como um conjunto de componentes inter-relacionados que colectam (recuperam), processam, armazenam e distribuem informações destinadas a tomada de decisões, a coordenação e ao controlo de uma organização (Laudon & Laudon, 2010).

Contudo (Azevedo, 2018) partilhando da mesma ideia defende que um SI consiste em um conjunto de mecanismos projectados com a finalidade de colectar, processar, armazenar e transmitir informações, de maneira a facilitar o acesso de utilizadores interessados, solucionando problemas e atendendo as suas necessidades, onde destaca a conversão de dados em informações como sendo o elemento essencial para a tomada de decisões mais assertivas.

Laudon e Laudon (2010) destacam também a existência de três (3) principais actividades que permitem gerar as conclusões que as organizações necessitam para o processo de tomada de decisão, a saber: a entrada, o processamento e a saída:

1. **Entrada** – que visa a colecta ou captura a de dados brutos da organização ou de um ambiente externo, que servirão de valores entrada para o processamento.
2. **Processamento** – que permite processar os dados brutos colectados ou capturados em uma forma mais significativa para que possa gerar resultados.
3. **Saída** – que resulta dos resultados gerados pelo **processamento**, transferindo as informações processadas a pessoas que utilizarão ou as actividades nas quais serão empregadas para que se possam efectuar tomadas de decisões mais assertivas.

Figura 1 - Processamento de dados para geração de informação (Fonte: autor)

## **2.2. Aplicação web**

Com o crescimento da tecnologia cada vez mais contaste na vida das pessoas e das organizações, torna-se quase que impossível gerir um negócio sem pensar no auxílio das Tecnologias de Informação (TI) para este processo, onde as aplicações web apresentam-se como uma das soluções mais viáveis, devido ao seu maior grau de acessibilidade.

Para (Macêdo, 2017), uma Aplicação web é um software que é instalado em um servidor web, onde é projetado para receber solicitações de clientes, processar e armazenar dados podendo dimensionar respostas de acordo de acordo com a demanda ou necessidade.

De um modo geral, as Aplicações web podem ser definidas como sendo um conjunto de Sistemas que são executados por meio da internet, onde baseiam-se nas três (3) principais atividades dos SI que englobam a entrada dados para Aplicação, o processamento dos dados recebidos e geração da saída de apoio a tomada de decisão, tendo em conta que este processo ocorre através de um navegador que funciona como uma camada intermediaria entre o utilizador e conteúdo disponível na internet (Tegra, 2020).

## **2.3. Protótipo**

Segundo (Sommerville, 2011), é uma versão inicial de *software* usada para demostrar conceitos, experimentar opções de projetos e descobrir mais sobre problemas e suas respectivas soluções, permitindo desta forma, melhor controlo de custos e a verificação da viabilidade da proposta, visto que o desenvolvimento de protótipos pode ajudar na identificação de erros e omissões nos requisitos propostos.

Para (Garimpo UX, 2018), protótipo é um modelo construído para testar um produto ou serviço na qual resulta de pesquisas iniciais relativas a uma ideia ou a suposições e, também como sendo uma base para que novas mudanças e implementações da ideia possam ser realizadas.

Deste modo, durante o processo de desenvolvimento é necessário que seja descrito o objetivo do protótipo para que possa ser compreendida a sua finalidade, que pode o ser o desenvolvimento de um protótipo para apresentar a interface do utilizador, a validação dos requisitos funcionais do aplicação ou a demonstração da viabilidade da aplicação.

### **2.3.1. Classificação de Protótipos**

De acordo com (Garimpo UX, 2018), os protótipos podem ser classificados quanto ao tipo e quanto a sua complexidade.

Quanto a sua complexidade podemos encontrar as seguintes classificações de protótipos:

* **Protótipo de Baixa fidelidade**

São utilizados no início do projeto para descartar incertezas, testar conceitos e descobrir valor para o produto, geralmente é feito a lápis e papel onde permitirá anotar as primeiras ideias.

Permite esboçar rapidamente como será o novo produto sem se preocupar com cores, aparência ou disposição dos elementos.

* **Protótipo de Média fidelidade**

Geralmente criados com programas de edição gráfica, estes protótipos tem maior apelo visual, entretanto, não possuem interações de tela e demandam mais tempo para se fazer ajustes e melhorias.

São uma ótima opção para telas com maior ênfase em estética e usabilidade, quando os requisitos já foram entendidos.

* **Protótipo de Alta fidelidade (Funcional)**

São protótipos completos, representativos e que mais se aproximam da realidade do produto e da experiência final, são geralmente utilizados para obter dados mais reais e significativos durante os testes de usabilidade e oferecem a possibilidade de observar pontos específicos como detalhes de estética e efeitos de interação.

## **2.4. *Unified Modeling Language* (UML)**

*Unified Modeling Language* (UML) pode ser entendida como sendo uma linguagem-padrão para a elaboração de projectos de *Software*, podendo ser empregada para a visualização, especificação, construção e documentação de artefactos de *Software*, ela permite abranger todas visões necessárias para o desenvolvimento e implementação de *Softwares*, provendo uma semântica que permita de certa forma capturar decisões estratégicas e tácticas independentemente das linguagens de programação e dos métodos de desenvolvimento a serem usados (Booch, Rumbaugh, & Jacobson, 2012).

Para (Ventura, 2019), UML é basicamente uma linguagem de notação para o uso em projectos de sistemas, em que em linhas gerais 50% do trabalho é a comunicação, compreendida como uma forma de escrever, ilustrar e comunicar através de diagramas, tendo em conta que cada diagrama é composto por elementos que possuem relação entre si, e os outros 50% trata-se da materialização daquilo que é combinado ou alinhado entre as partes envolvidas no processo.

### **2.4.1. Diagramas da UML**

Segundo (Booch, Rumbaugh, & Jacobson, 2012), um diagrama UML é uma representação em forma de gráficos onde são desenhados para permitir visualizar sistemas sob diversas perspectivas, na qual dentre os 13 diagramas da UML, para o presente trabalho serão usados os seguintes:

#### **2.4.1.1. Diagrama de Casos de uso**

Segundo (Viera, 2015), Diagrama de casos de uso permite auxiliar no processo de levantamento dos requisitos funcionais do Sistema, descrevendo um conjunto de funcionalidades do sistema e suas interacções com elementos externos e entre si, auxiliando de certa forma na comunicação entre o cliente e os analistas.

Para (Tacla), Diagramas de casos de uso são utilizados para representar de forma panorâmica os requisitos funcionais de um sistema do posto de vista do usuário, onde apresentam o comportamento através dos seus atores, casos de uso e relacionamentos.

##### **Actores**

Em diagramas de casos de usos, actores representam papéis desempenhados por utilizadores ou por qualquer outra entidade externa do sistema, na qual podem iniciar casos de uso bem como prover e/ou receber informações dos casos de uso.

##### **Casos de Uso**

Casos de uso representam todos os modos de execução do Sistema do ponto de vista do utilizador, isto é, um caso de uso pode ser entendido como sendo uma sequência de acções que produzem um resultado significativo para um determinado actor, onde são determinadas quais tarefas podem ser realizadas por um actor.

##### **Relacionamentos**

Relacionamentos permitem auxiliar na descrição dos casos de usos, onde destacam-se os seguintes tipos de relacionamentos:

1. **Associação** – que permite representar a interacção entre um actor e um caso de uso por meio de mensagens, é representado por uma linha sólida, com ou sem direcção.
2. **Inclusão** – geralmente utilizada em dois casos de uso, onde durante a execução de um caso de uso um outro caso é incluído obrigatoriamente.
3. **Extensão** – geralmente utilizada quando se deseja modelar um relacionamento alternativo, isto é, um caso pode estender outro quando se deseja inserir um comportamento opcional disparado por alguma condição.
4. **Generalização** - permite especificar comportamentos genéricos que podem ser especializados para atenderem necessidades específicas, onde atores apresentam responsabilidades ou características comuns.

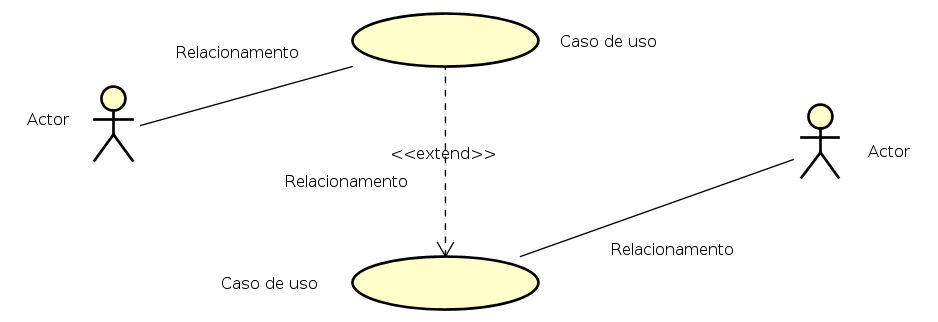


Figura 2 - Exemplo de representação de diagrama de classe (Fonte: Autor)

#### **2.4.1.2. Diagrama de Classes**

Diagrama de classes constituem um dos principais diagramas usados para desenvolver projectos de *Software* orientados a objecto, a sua especificação permite apresentar um conjunto de classes, interfaces e os seus respectivos relacionamentos, onde o requerido comportamento do sistema é alcançado pela colaboração entre objectos por meio dos seus respectivos relacionamentos, que podem ser: associação, agregação e generalização (Booch, Rumbaugh, & Jacobson, 2012).

##### **Relacionamentos em diagramas de classes**

###### **Associação**

Uma associação é um relacionamento estrutural que especifica a conexão ou associação entre o objecto de um item e objecto de outro item, isto é, de acordo com (Tacla), associação é uma relação entre duas classes significando que objectos destas possuem uma ligação, onde é possível navegar do objecto de uma classe até ao objecto de outra classe, geralmente é representado por uma linha sólida.

###### **Agregação**

Agregação constitui um caso especial de associação, utilizada para representar relacionamentos de pertinência, permitem representar que um objecto ou mais objectos de outras classes façam parte de um objecto de outra classe, segundo (Booch, Rumbaugh, & Jacobson, 2012, p. 122) “agregação é um tipo especial de associação, representando um relacionamento estrutural entre o todo e suas partes”.

###### **Composição**

Composição é uma agregação de fato, o todo é composto pelas partes, existe uma relação forte entre o todo e as partes, pois quando o todo é destruído as partes também o serão, ou seja, a eliminação do todo se propaga para as partes, de outra forma, o todo e as partes têm tempos de vida semelhantes.

Para (Booch, Rumbaugh, & Jacobson, 2012) , composição é uma forma de agregação, com propriedade bem definida e tempo de vida coincidente como parte do todo, um objecto será criado contendo as outras partes, onde uma vez criadas, vivem e morrem com ela.

###### **Generalização**

A relação de generalização é utilizada quando um conjunto de classes compartilha comportamentos e atributos, pode-se então generalizá-las agrupando seus comportamentos e atributos comuns, para (Booch, Rumbaugh, & Jacobson, 2012, p. 117) “uma generalizaçãoé um relacionamento entre itens gerais (chamados superclasses ou classes-mãe) e tipos mais específicos desses itens (chamados subclasses ou classes-filha)”.

##### **Componentes básicos de um diagrama de classes**

Um diagrama de classe é composto por três (3) partes, a saber:

* **Parte superior** – onde deve ser especificado o nome da classe, esta parte é sempre necessária para que se possa classificar que tipo de objecto a classe representa.
  + Uma **classe** é um *template* que permite representar as características e comportamentos de um determinado objecto.
* **Parte central** – onde são especificados os atributos/características das classes.
  + Um **atributo** é uma propriedade que permite descrever as características de um objecto representado em uma classe.
* **Parte inferior** – Onde são especificados a habilidades/operações (métodos) da classe.

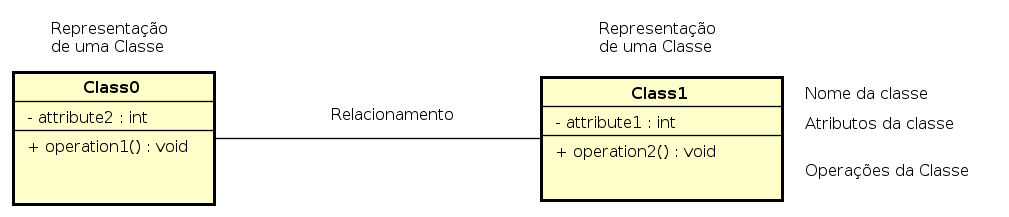


Figura 3 - Exemplo de representação de diagrama de classe (Fonte: Autor)

#### **2.4.1.3. Diagrama de Entidade-Relacionamento (DER)**

Segundo (Wixom & Dennis, 2014), Diagrama de Entidade e Relacionamento (DER) é um conjunto de figuras que mostram as informações criadas, armazenadas e usadas pelo sistema de um negócio. Este diagrama é muito semelhante com o diagrama de classe, todavia, no DER as informações do mesmo tipo ou grupo são agrupadas e colocadas no mesmo quadro denominado entidade, e os seus relacionamentos são estabelecidas por linhas entre os quadros junto com símbolos especiais que transmitem regra de negócio, diferenciando-se do diagrama de classe por este apresentar métodos e sua finalidade.

Um diagrama de entidade e relacionamentos pode ser composto por, entidades, relacionamento e atributos.

* **Entidade** – que pode ser entendido como algo que pode ser definido e que pode ter dados armazenados sobre ele, na qual pode ser identificada de forma inequívoca. Em um DER podemos ter os seguintes tipos de entidades:
  + **Entidade** **forte** – que pode ser definida unicamente pelos seus atributos.
  + **Entidade** **fraca** – onde possui existência dependente, ou seja, somente existe se existir a entidade da qual possuem parte dos atributos identificados.
  + **Entidade associativa** – que permite juntar entidades (ou elementos) dentro de um conjunto de entidades.
* **Relacionamento** – que pode ser entendido como sendo a forma como as entidades atuam uma sobre as outras. Em um DER podem ser definidos os seguintes tipos de relacionamento:
  + **Relacionamento unário ou Relacionamento Reflexivo** – que pode ser definida como sendo a situação em que uma entidade mantém relacionamento consigo mesma.
  + **Relacionamento binário** – que pode ser definida como sendo a situação em que uma entidade mantém relacionamento com uma outra entidade.
  + Relacionamento ternário – que compreende relações mais complexas, onde são definidos relacionamentos onde envolvem simultaneamente três (3) entidades.
* **Atributo** – é uma qualidade descritiva de uma entidade que assume valores distintos dentro de um domínio, pode ser definida também como sendo uma propriedade ou característica específica de uma entidade.

#### **2.4.1.4. Diagrama de actividades**

Diagrama de actividades é geralmente usada na fase de inicialização para descrever os fluxos de casos de uso, representação da dinâmica de negócio ou até mesmo para representar operações das classes, de acordo com (Booch, Rumbaugh, & Jacobson, 2012, p. 402) “os diagramas de actividades são um dos cinco diagramas disponíveis na UML para a modelagem de aspectos dinâmicos de sistemas. Um diagrama de actividade é essencialmente um gráfico de fluxo, mostrando o fluxo de controlo de uma actividade para outra”.

Em um diagrama de actividades possui os seguintes componentes básicos: Acção, actividade, transição, decisão, paralelismo ou bifurcação e sincronismo ou união.

##### **Componentes básicos**

* **Estado inicial** – que compreende o primeiro elemento do diagrama de actividades que indica o início do fluxo das actividades.
* **Acções** – que corresponde a uma etapa do diagrama de actividades em que o utilizador ou software realiza uma determinada tarefa.
* **Transição ou fluxo** – que corresponde a setas contínuas que representam o fluxo de trabalho de uma actividade para outra, isto é, o caminho a ser seguido até a conclusão do processo.
* **Decisões** – que corresponde ao controlo de desvio de fluxo, mediante a verificação de uma condição lógica.
* **Paralelismo ou bifurcação** – que compreende a representação de actividades concorrentes, em que resulta na divisão de fluxos de trabalho de uma actividade.
* **Sincronismo ou união** – que compreende a representação da união das actividades concorrentes, isto é, resulta na união fluxos de trabalho.
* **Raias** – que compreende a demonstração de como as actividades podem ocorrer especificando os agentes ou grupos responsáveis por essas actividades.
* **Estado final** – é o último elemento do diagrama que indica o fim do fluxo das actividades.

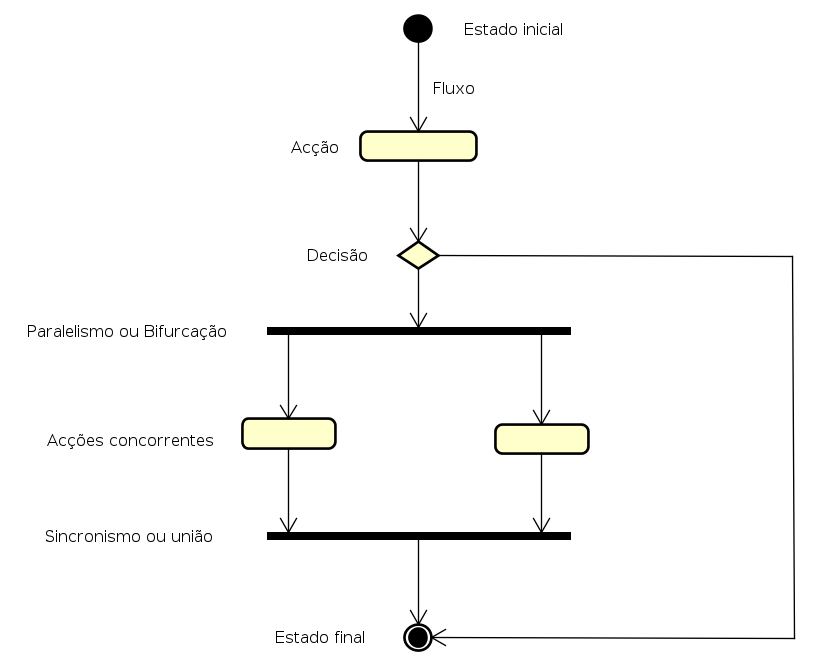


Figura 4 - Exemplo de representação de Diagrama de Actividades (Fonte: Autor)

#### **2.4.1.5. Diagrama de sequencias de eventos**

Segundo (Guedes, 2011), é um diagrama comportamental que preocupa-se com a ordem temporal em que as mensagens serão trocadas entre os objetos envolvidos em um determinado processo, na qual baseia-se em casos de uso definidos para aplicação. Podem apoiar-se em diagramas de classes para a identificação dos objetos das classes que fazem parte do processo.

Para (Cunha & Serafini, 2011), Diagrama de sequencia de eventos é uma representação gráfica e comportamental dos casos de uso da aplicação, na qual representa por meio de figuras um possível comportamento que a aplicação poderá ter.

A construção do diagrama de sequencia de eventos é dependente da definição dos casos de uso, isto é, enquanto os Diagramas de casos de uso preocupam-se em definir o papel das aplicações, os diagramas de sequencia de eventos preocupam-se em definir como a aplicação ira realizar um determinado papel, representando a forma como os grupos de objetos colaboram em um processo ao longo do tempo.

##### **Componentes básicos de um diagrama de sequencia de eventos**

Um diagrama de sequencia de eventos é composto por:

###### **Actores**

São os utilizadores que iniciam a interação e a troca de mensagens, isto é, são entidades externas que interagem com o sistema e que solicitam serviços, onde normalmente, o actor primário é o responsável por enviar a mensagem inicial que permitirá iniciar a interação entre os objetos.

###### **Objectos**

Representam as instâncias das classes representadas no processo.

###### **Linha de vida**

###### São linhas verticais que representam o tempo de vida de um objeto, um **X** na linha denota o fim da existência do objeto, por outro lado, os objetos no topo do diagrama são considerados criados desde o início.

###### **Fragmento**

É uma componente usado nos diagramas de sequência para destacar estruturas condicionais denotadas por Alt que equivale ao *if-else*, Opt que equivale ao *if*-sem else, e estruturas de repetição denotadas por *Loops* que equivalem ao *for-while*.

###### **Mensagem**

É utilizada para demonstrar a ocorrência de eventos, que normalmente forçam a chamada de um método em alguns objetos envolvidos no processo, as mensagens podem ser:

* **Simples** - quando o tipo de mensagem é irrelevante ou ainda não foi definida.
* **Síncrona** - quando enviada, o emissor fica bloqueado aguardando a resposta.
* **Assíncrona** – quando o emissor não fica bloqueado até o receptor enviar uma resposta para continuar o seu processamento.

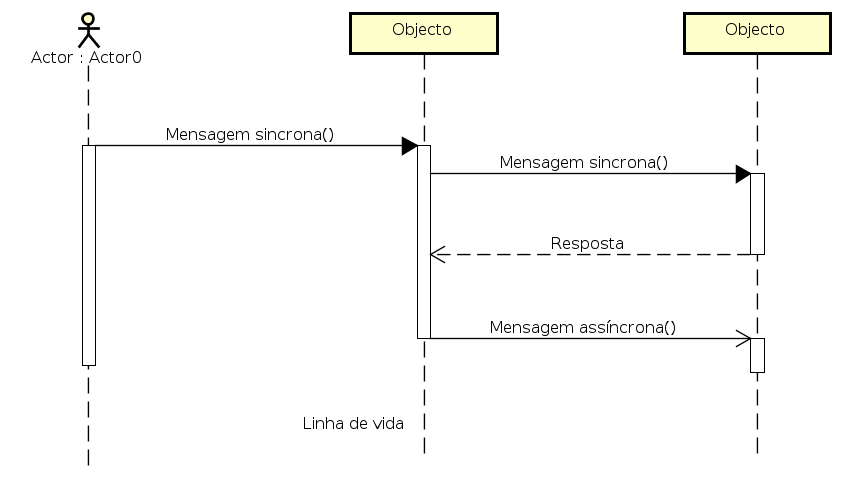
****

Figura 5 - Exemplo de Diagrama de sequência de eventos(Fonte: Autor)

# **METODOLOGIA**

De acordo com (Pradanov & Freitas, 2013) a pesquisa científica depende de um conjunto de procedimentos intelectuais e técnicos para que os seus objectivos sejam atingidos, por outro lado, um método científico pode ser visto como sendo um conjunto de processos ou operações mentais que devemos empregar na investigação.

Portanto, com vista a obtenção maior veracidade no conhecimento do problema bem como no alcance dos objectivos estabelecidos, o presente trabalho foi conduzido de acordo com os seguintes métodos científicos:

## **3.1. Metodologia de Pesquisa**

### **3.1.1. Quanto a natureza**

De acordo com (Pradanov & Freitas, 2013) uma pesquisa científica quanto a natureza pode ser classificada como básica ou aplicada. A pesquisa é classificada como básica quando envolve verdades universais, procurando gerar conhecimentos novos úteis para o avanço da ciência. E classificada como aplicada, quando visa produzir conhecimentos para aplicação pratica dirigidos a solução de problemas específicos.

Deste modo, tendo em conta a problemática levantada para o presente trabalho, quanto a natureza foi usada a pesquisa aplicada, com vista a prever mecanismos para solucionar as inconveniências que o processo de assinatura de livro de ponto para funcionários a tempo inteiro da USTM apresenta.

### **3.1.2. Quanto ao procedimento técnico**

Quando ao procedimento técnico, o presente trabalho foi conduzido de acordo com a pesquisa bibliográfica, onde segundo (Pradanov & Freitas, 2013), a pesquisa bibliográfica é definida como sendo aquela em que é concebida a partir de materiais já publicados, o que constitui grande parte deste trabalho, tendo sido analisados, livros e artigos científicos já publicados.

### **3.1.3. Quanto a abordagem**

Quanto a abordagem, uma pesquisa pode ser classificada como quantitativa ou qualitativa. A abordagem quantitativa, requer o uso de recursos e técnicas de estatísticas, procurando traduzir em números os conhecimentos gerados pelo pesquisador.

Tendo em conta o problema levantado, como forma de abordagem para este projecto foi conduzida a pesquisa qualitativa.

Na abordagem qualitativa o cientista objectiva aprofundar-se na compreensão dos fenómenos que estuda, as acções dos indivíduos, grupos ou organizações em seu ambiente ou contexto social, interpretando-os segundo a perspectiva dos próprios sujeitos que participam da situação, sem se preocupar com representação numérica, generalizações estatísticas e reacções lineares de causa e feito (Guerra, 2014, p. 11)

Partindo do exposto acima, a abordagem qualitativa adequa-se para esta pesquisa pois permitirá recolher dados de situações vivenciadas pelos funcionários da USTM, em relação aos problemas enfrentados para a assinatura do livro de ponto, desde as dificuldades até melhorias sugeridas.

### **3.1.4. População e amostra**

Para o presente projecto foi considerado como sendo o universo para o caso de estudo, todos os funcionários a tempo inteiro da USTM.

Segundo Marconi e Lakatos “A amostra é uma parcela convenientemente seleccionada do universo (População); é um subconjunto do universo” (2003, p. 163), por tanto será feita uma selecção de 20 funcionários da USTM para participar da pesquisa.

### **3.1.5. Técnica de recolha de dados**

Quanto aos procedimentos técnicos para a recolha de dados será usada a técnica da observação que consistirá na observação do processo de gestão do livro de ponto possibilitando a recolha de dados para a pesquisa desde a assinatura da entrada de funcionários até a sua saída.

Em adição, será usada também a técnica do questionário que será feito aos 20 funcionários que serão seleccionados para compor a amostra sendo que o questionário será apresentado em forma de formulário para o preenchimento.

Por fim, será feita uma análise de artigos já publicados assim como livros e teses, que serão de grande importância sobre tudo para ajudar na compreensão de opiniões e conclusões de estudos já feitos.

## **3.2. Metodologia de Desenvolvimento**

### **3.2.1.** ***Integrated Development Environment* (IDE)**

Segundo (Andrade, 2020), *Integrated Development Environment* (IDE) é definido como sendo um *Software* que auxilia no processo de desenvolvimento de aplicações, com objectivo de facilitar diversos processos ligados ao desenvolvimento de aplicações, combinando ferramentas comuns em uma única interface gráfica.

Deste modo, a IDE eleita para o presente trabalho é o *Spring Tool Suite* (STS), onde de acordo com (Afonso, 2017) é uma IDE baseada em *Eclipse* personalizada para o desenvolvimento de aplicações baseadas em projectos *Spring*, fornecendo um ambiente pronto para implementar, depurar, executar e implantar aplicativos **Spring.**

### **3.2.2. Modelagem do Sistema**

Conforme explica (Sommerville, 2011), Modelagem de Sistemas compreende ao processo de desenvolvimento de modelos abstractos de um sistema, onde cada modelo apresenta uma visão ou perspectiva diferente do Sistema, geralmente representado seguindo os padrões da UML, onde acredita-se que apenas 5 diagramas são necessários para representar a essência de um sistema, a saber: o diagrama de actividades, casos de uso, sequencia, classe e estado.

Para este estudo foi eleito a ferramenta de modelagem *Astah Community*, onde de acordo com (Lima, 2016) é um software para a modelagem UML, em que disponibiliza para desenvolvimento, os diagramas de Classes, Casos de Uso, Sequência, Estados, Actividade, e outros diagramas da UML.

### **3.2.3. Linguagem de programação**

Para (Gotardo, 2015) uma linguagem de programação é um método padronizado que é usado para expressar instruções de um programa a um computador programável, onde de certa forma torna-se possível saber quais dados serão usados, como esses dados serão processados, transumidos e armazenados, bem como quais acções devem ser tomadas em determinadas circunstâncias.

Para este estudo foi eleita a linguagem de programação Java, que apresenta-se como uma linguagem computacional completa para desenvolvimento de aplicações baseadas na *internet*, redes internas, em que é caracterizada como robusta, segura, portável e com alto poder de processamento e desempenho.

### **3.2.4. Sistema de Gestão de Banco de dados**

Para (Elmasri & Navathe, 2005), Um Sistema de Gestão de Banco de Dados (SGBD) é um colecção de programas que permite aos utilizadores criarem e manterem banco de dados, isto é possui o propósito geral de permitir com que os utilizadores facilidades nos processos de definição, manipulação e compartilhamento de banco de dados entre utilizadores e aplicações.

Para este estudo foi eleito de SGBD *MySQL*, em que de acordo com Milani (2006, p. 22) “O *MySQL* é um servidor e gerenciador de banco de dados (SGBD) relacional, projectado inicialmente para trabalhar em aplicações de pequeno e médio portes, mas hoje atendendo aplicações de grande porte e com mais vantagens do que os seus concorrentes”.

### **3.2.5. Sistema de controlo de versão**

Sistemas de controlo de versão possuem um papel importante no processo de desenvolvimento de *Software*, permitindo gerenciar versões no desenvolvimento de qualquer documento, porém, apresentam grande usabilidade no processo de desenvolvimento de *Software* para controlo de versões, históricos de desenvolvimento e documentação.

Conforme explicam (Chacon & Straub, 2014) um sistema de controlo de versão é um sistema que permite registar alterações de versões em um determinado arquivo ou conjunto de arquivos ao longo do tempo para que seja possível efectuar recuperações de versões especificas mais tarde de acordo com as necessidades.

Para o presente estudo, foi eleito *Git* como sistema de controlo de versão, por ser uma das soluções mais usadas, segura e por apresentar fácil integração com diversos mecanismos de hospedagem de código fonte como é o caso do *GitHub*.

# **CONCLUSÕES E RECOMENDAÇÕES**

## **4.1. Conclusões**

## **4.2. Recomendações**

# **REFERENCIAS BIBLIOGRÁFICAS**

Afonso, A. (2 de Fevereiro de 2017). *O que é Spring Boot?* Acesso em 19 de Agosto de 2020, disponível em Algaworks: https://blog.algaworks.com/spring-boot/

Andrade, A. P. (27 de Abril de 2020). *O que é uma IDE (Ambiente de Desenvolvimento Integrado)?* Acesso em 18 de Agosto de 2020, disponível em Treinaweb: https://www.treinaweb.com.br/blog/o-que-e-uma-ide-ambiente-de-desenvolvimento-integrado/

Azevedo, F. (10 de Maio de 2018). *O que é Sistemas de Informação?* Obtido em 25 de Junho de 2020, de Universidade Unigranio: https://portal.unigranrio.edu.br/blog/o-que-e-sistemas-de-informacao

Booch, G., Rumbaugh, J., & Jacobson, I. (2012). *UML: guia do usuário.* (F. F. Silva, & C. d. Machado, Trads.) Rio de Janeiro: Elsevier.

Caldeira, C. P. (2011). *Introdução aos Sistemas de Gestão de Informação.* Évora: Universidade de Évora.

Chacon, S., & Straub, B. (2014). *Pro Git* (2 ed.). New York: Apress.

Cunha, L. E., & Serafini, J. I. (2011). *Análise de Sistemas* (2 ed.). Colatina.

Elmasri, R., & Navathe, S. B. (2005). *Sistemas de Bancos de Dados.* São Paulo: Pearson.

Garimpo UX. (25 de Setembro de 2018). *QUAIS SÃO OS TIPOS DE PROTÓTIPO E O USO DE CADA UM*. Acesso em 3 de Outubro de 2020, disponível em Medium: https://medium.com/@garimpoux/quais-são-os-tipos-de-protótipo-e-o-uso-de-cada-um-dc6dfe4d10a0

Gotardo, R. A. (2015). *Linguagem de Programação I* (1 ed.). Rio de Janeiro: SESES.

Guedes, G. T. (2011). *UML 2: uma abordagem prática* (2 ed.). São Paulo: Novatec.

Guerra, E. L. (2014). *Manual de pesquisa Qualitativa.* Belo Horizonte.

Lakatos, E. M., & Marconi, M. d. (2003). *Fundamentos de metodologia cientifica.* São Paulo: Atlas.

Laudon, K. C., & Laudon, J. P. (2010). *Sistemas de Informação Gerenciais* (9 ed.). São Paulo: Pearson.

Lima, D. D. (20 de Julho de 2016). *Modele softwares com Astah Community*. Acesso em 29 de Julho de 2020, disponível em Techtudo: https://www.techtudo.com.br/tudo-sobre/astah-commmunity.html

Macêdo, D. (24 de Maio de 2017). *Entendendo as aplicações Web*. Acesso em 04 de Setembro de 2020, disponível em Diego Macêdo: Um pouco de tudo sobre T.I.: https://www.diegomacedo.com.br/entendendo-as-aplicacoes-web/

Milani, A. (2006). *MySQL : Guia do programador* (1 ed.). São Paulo: Novatec.

Pradanov, C. C., & Freitas, E. C. (2013). *Métodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico* (2 ed.). Novo Hamburgo - Rio Grande do Sul - Brasil: Feevale.

Santos, C. (25 de Setembro de 2017). *A Importância dos Colaboradores para o Futuro dos Negóci*. Obtido em 04 de Outubro de 2020, de Super Empreendedores: https://www.superempreendedores.com/empreendedorismo/recursos-humanos/importancia-dos-colaboradores/

Sommerville, I. (2011). *Engenharia de Software* (9 ed.). São Paulo: Person Prentice Hall.

Tacla, C. A. (s.d.). *Análise e projecto OO & UML 2.0.* Universidade Tecnológica Federal do Paraná.

Tegra. (25 de Março de 2020). *Aplicações web: o que são e quais suas vantagens? Entenda aqui*. Acesso em 4 de Outubro de 2020, disponível em Tegra: https://tegra.com.br/aplicacoes-web/

Ventura, P. (31 de Janeiro de 2019). *O que é UML (Unified Modeling Language)*. Obtido em 25 de Junho de 2020, de Até o Momento: https://ateomomento.com.br/diagramas-uml

Viera, R. (12 de Dezembro de 2015). *UML - Diagrama de Casos de Uso*. Acesso em 20 de Junho de 2020, disponível em OperacionalTI: https://medium.com/operacionalti/uml-diagrama-de-casos-de-uso-29f4358ce4d5

Wixom, B. H., & Dennis, A. (2014). *Análise e Projeto de Sistemas* (5 ed.). Rio de Janeiro: LTC.