**UML**

Antes de iniciar a codificação de um sistema, é necessário modelar ele, portanto, é nesse processo que utilizamos a UML (Unified Modeling Language ou Linguagem de Modelagem Unificada).

Deve-se ressaltar que UML não é uma linguagem de programação, e sim, uma linguagem visual que através de diagramas se torna possível compreender melhor o produto final, além de minimizar possíveis erros futuros. A indústria internacional de engenharia de software adota essa linguagem de modelagem como padrão, pois é o melhor modelo disponível no mercado.

**Por que modelar um sistema?**

Existe uma diferença gritante entre construir uma pequena casa e construir um prédio de vários andares. Obviamente, para se construir um edifício, é necessário, em primeiro lugar, desenvolver um projeto muito bem elaborado, cujos cálculos têm de estar corretos e precisos. Além disso, é preciso fornecer uma estimativa de custos, determinar em quanto tempo a construção estará concluída, avaliar a quantidade de profissionais necessária à execução da obra, especificar a quantidade de material a ser adquirida para a construção, escolher o local onde o prédio será erguido etc. Grandes projetos não podem ser modelados de cabeça, nem mesmo a maioria dos pequenos projetos pode sê-lo, exceto, talvez, aqueles extremamente simples. (GUEDES, 2011)

**Análise de Requisitos**

Antes de modelar um programa, é preciso levantar seus requisitos e analisa-los, isto é, entender e documentar o que o cliente deseja para o seu software. Os requisitos são divididos em três partes: Requisito Funcional (RF), Requisito Não Funcional (RNF) e Regra de Negócio (RN).

* Requisitos Funcionais são todas as funcionalidades principais de um sistema, aquilo que se você excluir, o software terá problemas de execução. Abaixo encontra-se os requisitos funcionais do sistema Epoint.

**RF01**: Disponibilizar sistema de login.

**RF02**: Pesquisar eventos.

**RF03**: Cadastrar eventos.

**RF03**: Disponibilizar tela para gerenciamento dos eventos.

**RF04**: Disponibilizar Inscrições nos eventos.

**RF05**: Disponibilizar método de Pagamentos.

**RF06**: Disponibilizar canal de comunicação entre organizador e participantes.

**RF07**: Avaliar eventos.

**RF08**: Conter tela de calendário dos eventos.

**RF09**: Conter Sistema de ranking.

**RF10**: Disponibilizar suporte técnico.

**RF11:** Disponibilizar método para excluir usuários por parte dos administradores

**RF12:** Disponibilizar método para excluir eventos por parte dos administradores

**RF14:** Cadastrar administradores

**RF15:** Cancelar inscrição do evento

**RF16:** Disponibilizar configuração de conta

* Requisitos não funcionais são funções do sistema que são importantes, no entanto, diferente dos funcionais, eles não devem ser obrigatórios para a execução do software. Entende-se como RNF: segurança, usabilidade, desempenho, entre outros. Em sua maioria, o levantamento desses requisitos deve ser feito pelo analista do projeto, respeitando a ideia do cliente. Abaixo é possível conferir os requisitos não funcionais do sistema Epoint.

**RNF01**: O site deve ser responsivo e rápido, para garantir que os usuários possam acessar as informações dos eventos rapidamente e sem interrupções.

**RNF02**: O site deve ser intuitivo e fácil de usar, para que o usuário usufrua da melhor forma possível.

**RNF03**: O site deve ser compatível com diferentes dispositivos e sistemas operacionais, permitindo que os usuários acessem o site de qualquer lugar e dispositivo.

**RNF04**: O site deve ser acessível para pessoas com deficiências visuais, motoras e auditivas, oferecendo opções de acessibilidade e recursos de navegação.

* Regras de negócio são idealizadas pelo cliente, tendo como objetivo estabelecer normas para a execução do sistema. Imagine um sistema cujo objetivo é o empréstimo de livros, a regra de negócio determinada pelo cliente seria um livro por pessoa para empréstimo, sendo assim, o sistema deve respeitar essa diretriz e bloquear caso o usuário queira pegar mais de um livro. Segue abaixo as regras de negócio do sistema Epoint.

**RN01**: O site deve permitir que os usuários se registrem para criar e gerenciar seus próprios eventos competitivos.

**RN02**: O site deve permitir que os usuários pesquisem eventos competitivos por data, local, tipo de evento, etc.

**RN03**: O site deve permitir que os usuários cadastrem novos eventos competitivos com informações relevantes, incluindo datas, localização, tipo de evento, categoria, preço de entrada, regras, prêmios, etc.

**RN04**: O site deve permitir que os organizadores gerenciem seus próprios eventos, incluindo atualizações, adição de informações e cancelamentos.

**RN05**: O site deve permitir que os usuários se inscrevam em eventos competitivos, fornecendo informações pessoais para o organizador como nome, idade, contato, etc.

**RN06**: O site deve permitir que os usuários paguem pelas inscrições em eventos competitivos, usando um sistema de pagamento seguro e confiável.

**RN07**: O site deve disponibilizar um chat, permitindo que o organizador e o inscrito se comuniquem entre si, para tirar dúvidas, resolver problemas, trocar informações, etc.

**RN08**: O site deve permitir que os usuários avaliem os eventos que participaram, para que outros usuários possam ter uma ideia da qualidade dos eventos ou do organizador.

**RN09**: O site deve ter um calendário com todas as datas dos eventos cadastrados, para facilitar a visualização dos eventos futuros e a organização dos usuários.

**RN10**: O site poderia ter um sistema de ranking para os eventos e participantes, baseado em critérios como pontuação, número de vitórias, nível de dificuldade, entre outros.

**RN11**: O site deve ter um suporte técnico disponível para os usuários, para ajudá-los a solucionar eventuais problemas técnicos, dúvidas ou sugestões de melhorias.

**RN12**: O site deve permitir que os usuários deem feedback sobre as funcionalidades e usabilidade do site, para que os desenvolvedores possam melhorá-lo continuamente e atender às necessidades dos usuários.

**RN13:** Após se inscrever, o participante receberá em seu e-mail as informações sobre o evento

**RN14:** Após se inscrever, o participante terá as informações sobre o evento em uma aba específica do site

**RN15:** A organização poderá ver as informações de quem se inscreveu no próprio site

**RN16:** Caso seja necessário pagar para participar do evento, a inscrição do participante só será confirmada após a oficialização do pagamento.

**RN17:** Não será possível se inscrever para dois eventos ou mais que estejam registrados na mesma hora e mesmo dia.

**RN18:** O site deve indicar eventos de acordo com a preferência do usuário.

**RN19:** A plataforma deve deixar claro e explícito que não organiza os eventos, portanto, caso ocorra algum problema, não é de responsabilidade da Epoint.

**RN20:** A organização poderá cancelar o evento a qualquer hora, sem aviso prévio.

**RN21:** O participante poderá cancelar sua participação até 30 minutos antes do início do evento.

**RN22:** Poderá ser cadastrado como evento no site as seguintes categorias: competições de esportes, e-sports e gincanas.

**RN23:** Qualquer evento que não seja das categorias especificadas não poderá ser registrado no site.

**Diagrama de Máquinas de Estado**

O diagrama de máquina de estados demonstra o comportamento de um elemento por meio de um conjunto finito de transições de estado. Além de ser utilizado para expressar o comportamento de uma parte do sistema [...] (GUEDES, 2011)

Antes de ser apresentado os diagramas de máquina de estados do sistema Epoint, é fundamental conhecer e entender as figuras que compõem esse diagrama, portanto, segue abaixo:

* **Estado Simples** é o estado no qual que se encontra o sistema, geralmente é chamado apenas de “estado”. 

Figura 1 – Exemplo de Estado Simples.

* **Transições** é um evento que causa mudança no estado.

Figura 2 – Exemplo de Transição.

* **Estado Inicial** tem como objetivo apenas de identificar a inicialização do diagrama. 

Figura 3 – Exemplo de Estado Inicial.

* **Estado Final** é usado em situações onde o programa pode ser finalizado.

Figura 4 – Exemplo de Estado Final.

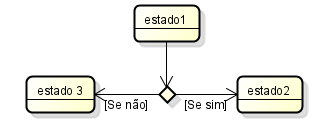
* **Pseudoestado de Escolha** é um evento onde o usuário precisa tomar uma decisão, após feita a escolha, o sistema é alterado para o respectivo estado. 

Figura 5 – Exemplo de Pseudoestado de Escolha.

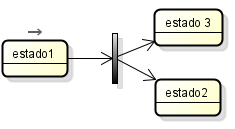
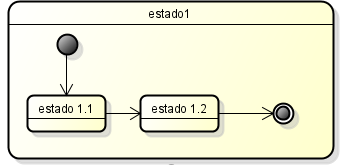
* **Barra de Bifurcação/União** serve para determinar quando ocorre estados paralelos e em quantos subprocessos de dividiu (bifurcação) ou especificar o momento que os subprocessos se uniram (união).

Figura 6 – Exemplo Barra de Bifurcação.

* **Estados Compostos** contém internamente dois ou mais subestados, detalhando melhor o processo do estado-mãe.

Figura 7 – Exemplo de Estado Composto.

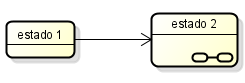
* **Estado de Submáquina** equivale ao estado composto, porém, os subestados não são descritos no diagrama, o que se faz necessário representa-los em outro. É uma função utilizada para desenhar diagramas mais enxutos, quando há projetos mais complexos. O Estado de Submáquina é identificado por um símbolo no canto inferior direito. 

Figura 8 – Exemplo de Estado de Submáquina.

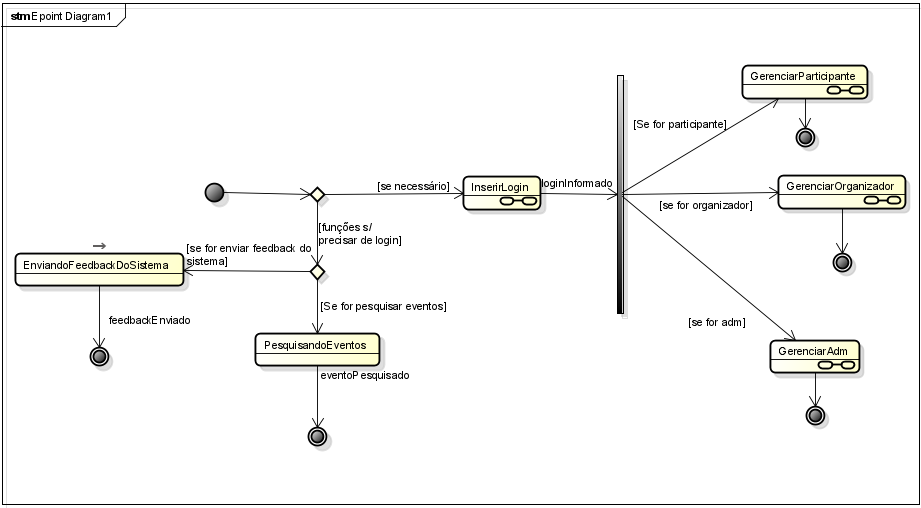
Com a apresentação dos elementos de um diagrama de máquina de estado, será apresentado na sequência os diagramas referentes ao sistema Epoint.

Figura 9 – Diagrama Máquina de Estado Sistema Epoint.

No diagrama fica explícito que o usuário consegue realizar duas funções sem precisar logar, são elas: enviar um feedback do sistema para os desenvolvedores do site e pesquisar os eventos desejados. É determinado que para se inscrever (participante), gerenciar seu evento (organizador) ou administrar o site (administrador), inserir o login se torna obrigatório. Caso o usuário ainda não tenha uma conta, é necessário fazer um cadastro, um processo que está detalhado no diagrama “Inserir Login”.

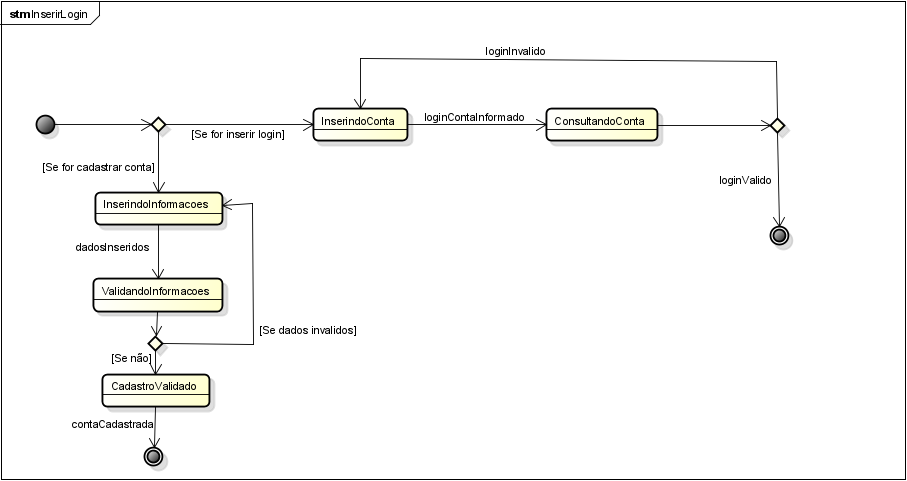
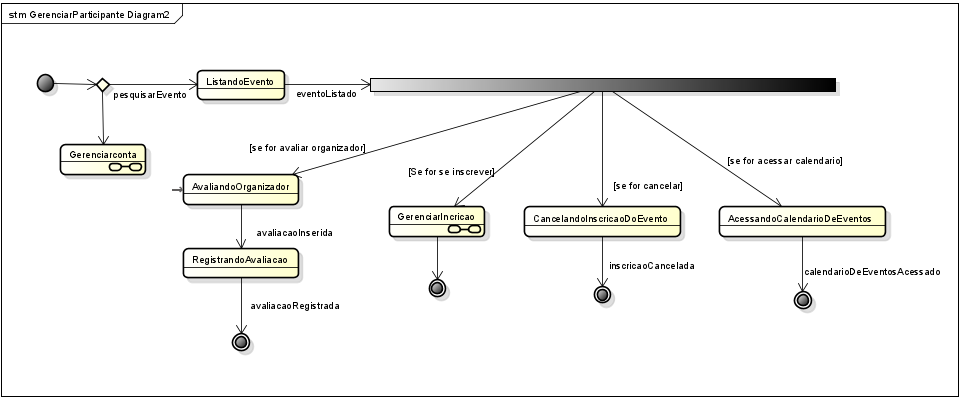


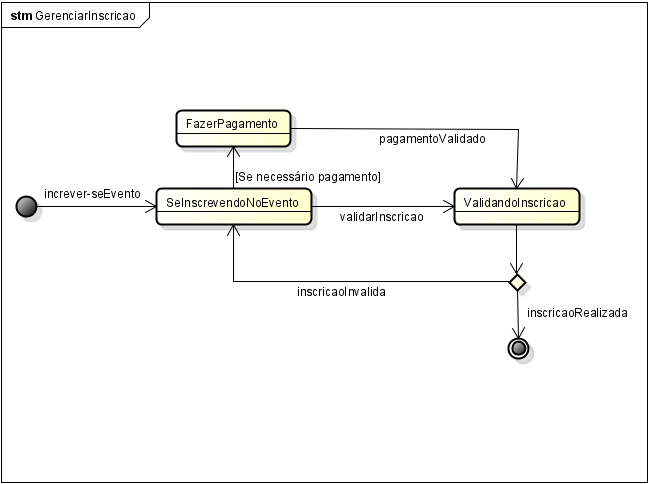
Figura 10 - Diagrama Máquina de Estado Inserir Login.

Nesse diagrama percebemos que o sistema oferece duas opções, o usuário pode inserir seu login e o sistema irá verificar se realmente existe a conta, ou, caso contrário, o usuário pode se cadastrar.

Com base no máquina de estado do sistema Epoint, segue o diagrama Gerenciar Participante.

Figura 11 – Diagrama Máquina de Estado Gerenciar Participante

Podemos perceber que a partir do momento em que o participante listar o evento, o sistema disponibiliza quatro opções: avalizar organizador, acessar o calendário dos eventos em que já participou ou participará, cancelar a participação de um evento, ou se inscrever no evento (processo relatado no próximo diagrama).

Figura 12 – Diagrama Máquina de Estado Gerenciar Inscrição.

Caso precise pagar para participar do evento, o sistema deve primeiro verificar se o pagamento foi validado, depois realizar a inscrição.

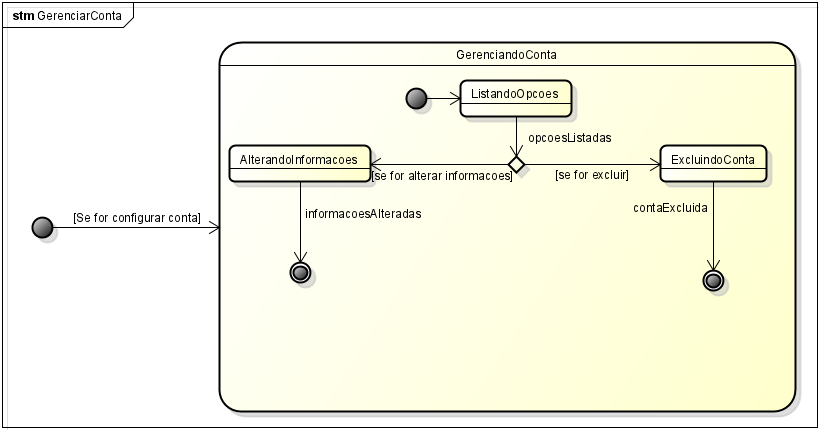
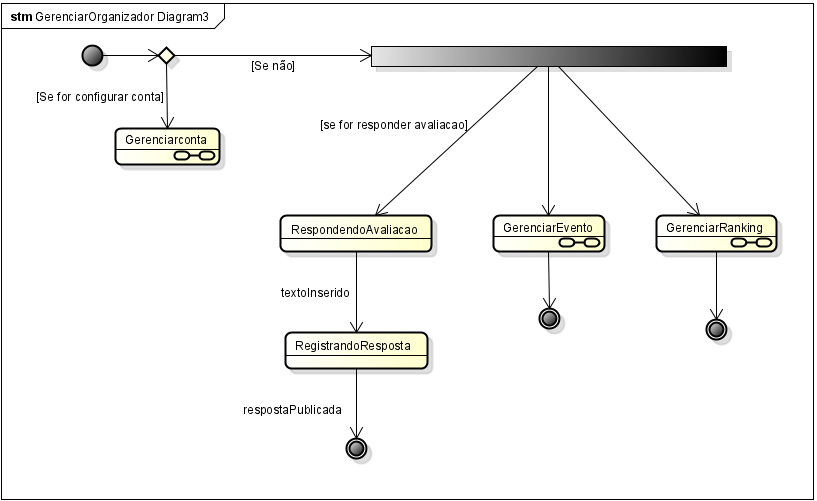
De acordo com o diagrama Gerenciar Participante, além de listar o evento, o mesmo consegue gerenciar sua conta. 

Figura 13 – Diagrama Máquina de Estado Gerenciar Conta.

Com base no diagrama do sistema Epoint, segue o diagrama Gerenciar Organizador.

Figura 14 – Diagrama Máquina de Estado Gerenciar Organizador.

Como o participante, o organizador também pode configurar sua conta, mas caso não deseje isso, ele pode responder as avaliações sobre o evento em questão, gerenciar seu evento ou gerenciar os rankings de determinado evento, como mostra os diagramas a seguir.

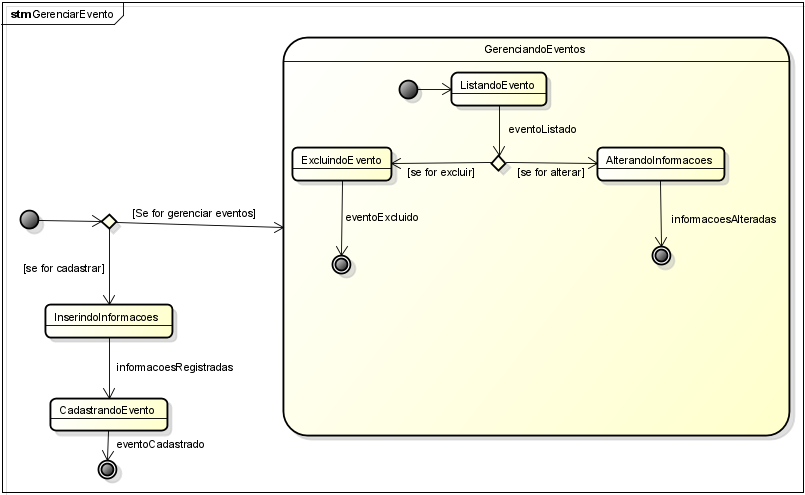
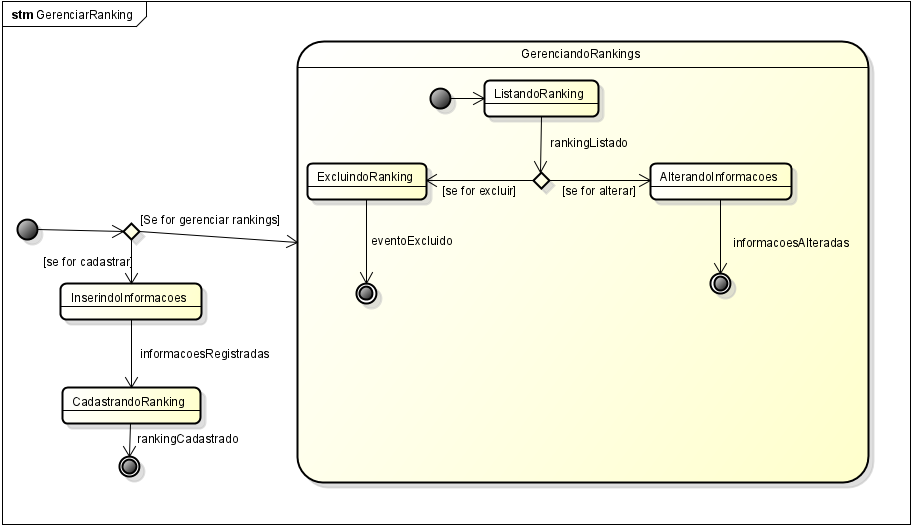
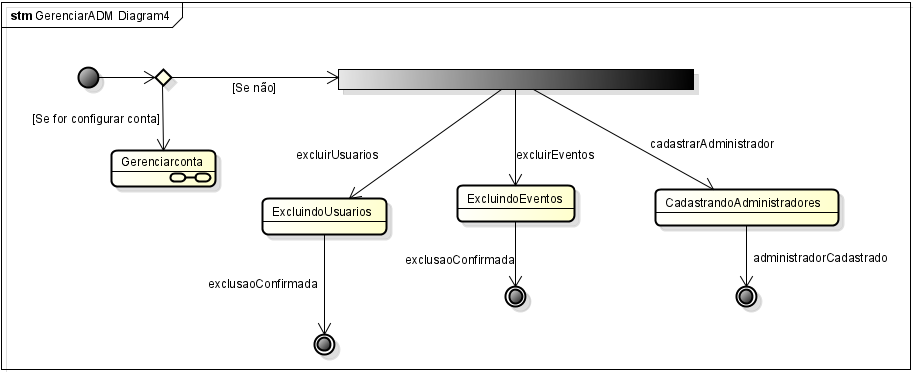
Figura 15 – Diagrama Máquina de Estado Gerenciar Evento.

Figura 16 – Diagrama Máquina de Estado Gerenciar Ranking.

Com base no diagrama do sistema Epoint, encontra-se a seguir o diagrama Gerenciar Administrador(ADM).

Figura 17 – Diagrama Máquina de Estado Gerenciar ADM.

Como os outros usuários, o adm pode gerenciar sua conta, além disso, tem como função excluir usuário que achar inadequado, excluir qualquer evento no qual acredita que não agrega valor, e por fim, há a opção de adicionar novos administradores.