



Universidade Federal da Paraíba

Centro de Informática

Métodos de Projeto de Software

# Facefy

*Playlist Colaborativa*

Caio Marcelo Campoy Guedes - 11228207

Marcelo Aguiar Rodrigues - 11311862

Wesnydy Lima Ribeiro - 11328275

# Glossário

**Group Social Network** - Rede social de interação entre pessoas. Essa rede social deve permitir a criação de eventos ou grupos. A lista de participantes desses eventos ou grupos deve poder ser capturada através de uma API.

**Music Social Network** - Serviço de streaming de música que tenha características de uma rede social de interação entre pessoas. Esse serviço deve fornecer as músicas e permitir a criação de playlists.

**Playlist** - Conjunto de músicas disponíveis na Music Social Network.

**Usuário** - São os clientes da aplicação. Eles utilizaram as funcionalidades de criação, edição e deleção das playlists.

**Evento** - É uma representação na Group Social Network de um evento de pessoas na vida real. Esse evento deve ter uma lista de participantes convidados, lista de participantes confirmados, horário e data.

**Grupo** - É uma representação na Group Social Network de um grupo de pessoas. Esse grupo deve ter uma lista de participantes.

**Participantes** - São as pessoas que confirmaram participação em um evento ou pessoas que fazem parte de um grupo.

# 1. Introdução

Este documento descreve as técnicas, bem como os requisitos e casos de uso utilizados para elicitar, analisar e especificar a construção do aplicativo Facefy. Aplicação que cria playlists colaborativas através do uso de APIs web.

## 1.1. Visão Geral do Problema

Encontrar músicas que satisfaçam o interesse de todos os participantes de um evento não é algo fácil. O Facefy é uma ideia resultante de tal problema. Porque não criar uma playlist com base no interesse de todos os participantes? Para tal o aplicativo busca unir as músicas favoritas dos usuários em uma playlist colaborativa. Tal playlist é criada a partir dos convidados confirmados para um evento numa rede social de amigos. Com essa lista em mãos será usada uma rede social de músicas para gerar uma playlist baseada nas preferências desses usuários.

## 1.2 Informações sobre o Projeto

### 1.2.1. Nome da aplicação

O nome escolhido para a aplicação é “Facefy”.

### 1.2.2. Descrição

O aplicativo busca unir as músicas favoritas dos usuários em uma playlist colaborativa. Tal playlist é criada através de duas APIs, a do Groups Social Network, da qual informações dos usuários serão coletadas a partir de um evento ou de um grupo. E a do Music Social Network que verifica as preferências dos usuários e cria uma playlist com base nestas.

### 1.2.3. Endereço URL

Tanto o endereço para navegadores desktop quanto para navegadores mobile será <http://www.facefy.com>.

### 1.2.4. Stakeholders

Os stakeholders da aplicação estão listados abaixo sem qualquer ordem de prioridade.

- Caio Guedes
- Marcelo Rodrigues
- Wesnydy Ribeiro

- Pessoas interessadas nos eventos do Groups Social Network
- Membros de grupos do Groups Social Network
- Criadores de eventos do Groups Social Network
- Administradores de grupos do Groups Social Network

## **2. Elicitação de Requisitos**

Para elicitar os requisitos do software foram utilizadas duas técnicas, o brainstorming e o questionário. Estas técnicas foram utilizadas por serem ágeis, uma vez que o tempo para elicitação teve de ser otimizado.

### **2.1. Brainstorming**

Foram realizadas três reuniões presenciais de brainstorming com o intuito de definir o escopo e as tecnologias que devem ser utilizadas para o desenvolvimento. As notas das reuniões podem ser encontradas abaixo.

#### **2.1.1 [BST01] - Primeira sessão de brainstorm.**

**Participantes:** Caio Guedes

Marcelo Rodrigues

Wesnydy Ribeiro

**Data:** 14/02/2017

Durante a sessão foi sugerido fazer algo com a Web API da Music Social Network. A ideia seria gerar uma playlist direcionada à um evento de forma colaborativa ao invés de uma única pessoa definir as músicas do evento. Surgiu então a ideia de fazer uma playlist baseada em um evento do Facebook. O dono da festa criaria um evento e os seus convidados e o sistema iria buscar a partir da lista de usuários confirmados quais tenham conta na Spotify e criar uma playlist com um mix contendo as preferências musicais de todos os que iriam comparecer ao evento.

#### **2.1.2. [BST02] - Segunda sessão de brainstorm.**

**Participantes:** Caio Guedes

Marcelo Rodrigues

Wesnydy Ribeiro

**Data:** 17/02/2017

Como ficou definido utilizar as APIs da Spotify e do Facebook fizemos uma pesquisa para saber a viabilidade da implementação. A API do Facebook permite resgatar a lista de convidados de um evento assim como a lista de pessoas de um grupo, então ficou decidido que seria possível criar playlists tanto para eventos esporádico quanto para grupos fixos do Facebook. A API da Spotify permite buscar apenas pelas playlists públicas de usuários não logados. Então para a implementação o que teríamos que fazer é buscar dentro das playlists públicas de cada usuário pelos artistas mais mencionados, que seriam os artistas mais ouvidos pelo usuário, para gerar uma lista de artistas. A partir dessa lista a playlist será criada com as músicas mais populares do Spotify.

### 2.1.3 [BST03] - Terceira sessão de brainstorm.

**Participantes:** Caio Guedes

Marcelo Rodrigues

Wesnydy Ribeiro

**Data:** 05/04/2017

Após uma pesquisa mais profunda sobre a viabilidade de utilizar as APIs do Facebook e do Spotify, foi detectado que algumas informações essenciais para a aplicação não podem ser obtidas, com isso, algumas abordagens decididas nas sessões anteriores se tornaram inviáveis para o desenvolvimento da aplicação. Nesta última sessão foi decidido fazer uma remodelagem na abordagem anterior para generalizar as redes sociais envolvidas na aplicação.

## 2.2. Questionário

Com essas ideias em mãos foi criado um questionário no Google Docs para investigar a aceitação da aplicação no mercado. As perguntas do questionário e as respostas estão listadas abaixo.

**Data:** 02/03/2017

**Local:** Laboratório de Aplicações de Vídeo Digital (LAViD) da UFPB

**Perfil dos usuários:** Desenvolvedores e colaboradores do LAViD

<b>Indicação de data e hora</b>	<b>Sabendo que o aplicativo é uma playlist colaborativa para Grupos e Eventos de Redes Sociais classifique de 1 a 5, o quão inclinado a usar o aplicativo você está.</b>	<b>O quão inclinado você está para utilizar a opção de criar playlists a partir de eventos? Classifique de 1 a 5.</b>	<b>O quão inclinado você está para utilizar a opção de criar playlists a partir de grupos? Classifique de 1 a 5.</b>	<b>Você compartilharia a aplicação com seus amigos?</b>	<b>Quer nos ajudar? Deixe uma sugestão.</b>
<b>02/03/2017 16:12:59</b>	3	5	3	Sim	
<b>02/03/2017 16:14:13</b>	4	3	3	Sim	
<b>02/03/2017 16:38:27</b>	5	5	5	Sim	Cria logo. Super interessante.
<b>02/03/2017 16:38:28</b>	5	5	5	Sim	Cria logo. Super interessante.
<b>02/03/2017 16:42:38</b>	5	5	3	Sim	
<b>02/03/2017 16:43:15</b>	3	1	3	Sim	
<b>02/03/2017 16:53:53</b>	3	3	5	Sim	
<b>02/03/2017 17:32:25</b>	5	5	5	Sim	

A partir desses levantamentos concluímos que existe demanda para um sistema nos moldes que pensamos. Um sistema que crie uma playlist em algum serviço de stream de músicas baseada nos participantes de um evento ou grupo específico numa rede social de amigos.

Apesar das respostas do questionário indicarem que alguns usuários não utilizariam a aplicação eles estariam fortemente inclinados à compartilharem a aplicação com outras pessoas.

## **2.3. Requisitos Funcionais**

A partir da elicitación foram levantados os seguintes requisitos funcionais para a aplicação.

### **2.3.1. [RF001] - Grupos e eventos**

O sistema deve criar playlists tanto de eventos quanto de grupos.

**Prioridade:** Deve existir

### **2.3.2. [RF002] - Identificadores**

O sistema deve associar um identificador em comum à uma playlist e um evento ou grupo.

**Prioridade:** Deve existir

### **2.3.3. [RF003] - Notificação**

O sistema deve informar ao usuário assim que a playlist colaborativa estiver concluída retornando para a listagem de playlist criadas na tela principal com a nova playlist destacada das demais.

**Prioridade:** Deve existir

### **2.3.4. [RF004] - Duração da playlist**

No momento de criação da playlist o usuário deve poder setar uma duração aproximada da playlist. A duração deve ser indicada em horas e minutos com intervalos de 30 minutos. O valor máximo para horas é de 05 horas e o mínimo de 30 minutos. Durações como 2h45m não devem ser aceitas.

**Prioridade:** Deve existir

### **2.3.5. [RF005] - Histórico**

O usuário pode buscar uma playlist criada anteriormente.

**Prioridade:** Recomendado

### **2.3.6. [RF006] - Relatório**

O sistema deve gerar relatórios estatísticos tanto em formato PDF quanto no formato HTML para o administrador.

**Prioridade:** Deve existir

### **2.3.7. [RF007] - Filtro de gênero**

O usuário pode filtrar as músicas a serem colocadas na playlist selecionando gêneros específicos.

**Prioridade:** Recomendado

### **2.3.8. [RF008] - Informações de colaboração**

O usuário pode buscar lista de participantes que colaboraram para a criação da playlist.

**Prioridade:** Opcional

### **2.3.9. [RF009] - Compartilhamento**

O usuário pode compartilhar a playlist em alguma rede social ou em algum aplicativo de troca mensagens.

**Prioridade:** Opcional

### **2.3.10. [RF010] - Confirmação de pessoas no evento**

Verificação de todas as pessoas que confirmaram presença no evento da rede social de pessoas.

**Prioridade:** Deve existir

### **2.3.11. [RF011] - Lista de membros**

Buscar a lista de membros do grupo da rede social pessoas.

**Prioridade:** Deve existir



### 2.3.12. [RF012] - Verificação de usuários cross-platform

Identificação de perfis que existem tanto na rede social de pessoas quanto na rede social de músicas.

**Prioridade:** Deve existir

## 2.4. Requisitos Não Funcionais

A partir da elicitación foram levantados os seguintes requisitos funcionais para a aplicação.

### 2.4.1. [RNF001] - Design

Interface do usuário implementada usando material design.

**Prioridade:** Deve existir

**Classificação:** Requisito de implementação

### 2.4.2. [RNF002] - Compatibilidade e Portabilidade

Compatibilidade com browsers desktop e mobile nas versões Google Chrome 57, Firefox 52, Safar 10.1i, Microsoft Edge 38, ou versões superiores.

**Prioridade:** Deve existir

**Classificação:** Requisito de portabilidade

### 2.4.3. [RNF003] - Responsividade

Interface autojustável à displays com resolução mínima 1920x1080 em modo landscape ou portrait.

**Prioridade:** Deve existir

**Classificação:** Requisito de produto

### 2.4.4. [RNF004] - Persistência

O sistema deve ser capaz de armazenar informações das playlists criadas, para isso deve se comunicar com o mongoDB

**Prioridade:** Deve existir

**Classificação:** Requisito de interoperabilidade

### **2.4.5. [RNF005] - Tempo de resposta**

Definição de timeout de 30 segundos para a aplicação ser executada e retornar a playlist gerada ao usuário

**Prioridade:** Deve existir

**Classificação:** Requisito de produto

### **2.4.6. [RNF006] - Front end e Framework**

A aplicação deve utilizar o framework bootstrap para desenvolver seu front end.

**Prioridade:** Deve existir

**Classificação:** Requisito de implementação

### **2.4.7. [RNF007] - Tecnologia para implementação**

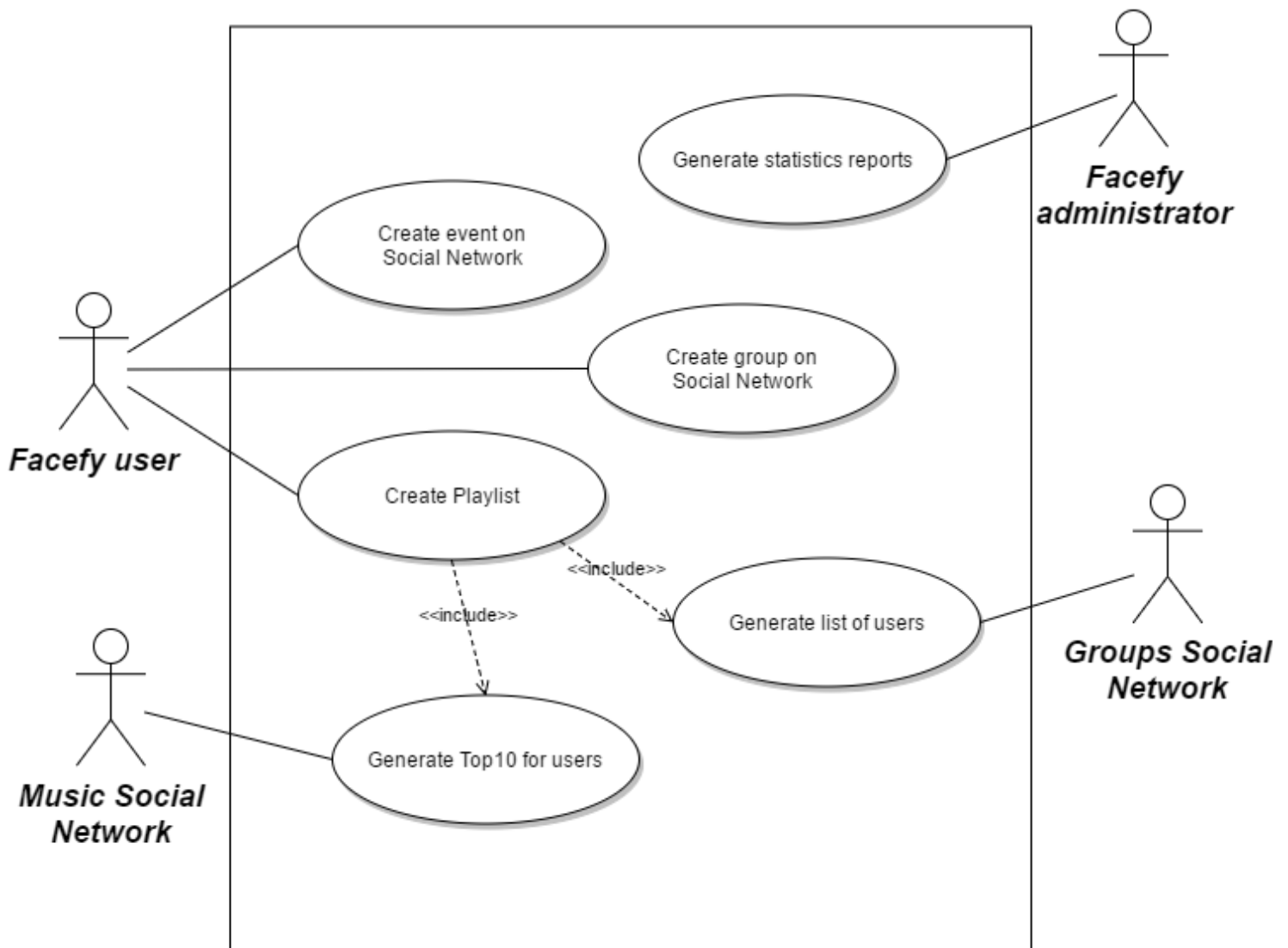
A aplicação deve utilizar Javascript com padrão EcmaScript6.

**Prioridade:** Deve existir

**Classificação:** Requisito de implementação

## 3. Casos de Uso

### 3.1. Diagrama de Casos de Uso



## 3.2. Descrição dos casos de uso

### 3.2.1. [UC01] Create Playlist

<b>Identificador</b>	[UC01]
<b>Nome</b>	Create Playlist
<b>Objetivo</b>	Criar uma playlist para um evento
<b>Requisitos</b>	RF01; RF02; RF03; RF04; RF05; RF13
<b>Atores</b>	Groups Social Network (GSN), Music Social Network (MSN), Facefy User
<b>Pré-condições</b>	O usuário deve estar logado no sistema, ter uma conta ativa e logada no GSN assim como um evento criado no GSN e uma conta ativa e logada na MSN.
<b>Frequência de uso</b>	Alta
<b>Pós-condições</b>	Deve existir uma playlist criada na conta do usuário na MSN onde as músicas foram escolhidas baseadas nas preferências das pessoas que participam do evento/grupo do GSN.
<b>Fluxo Principal [FP]</b>	<ol style="list-style-type: none"> <li>1. Usuário indica um evento/grupo do GSN ao sistema.</li> <li>2. O sistema requisita do GSN a lista de pessoas do evento/grupo. [FS01] [FS02]</li> <li>3. O GSN fornece a lista de pessoas do evento/grupo.</li> <li>4. &lt;&lt;include&gt;&gt; [UC002].</li> <li>5. O usuário define a duração da playlist.</li> <li>6. O sistema solicita a criação de uma playlist na conta do usuário na MSN. [FS02] [FS03]</li> <li>7. A MSN cria a playlist.</li> <li>8. O sistema adiciona a lista de músicas na playlist.</li> </ol>
<b>Fluxo Secundário [FS01]</b>	<ol style="list-style-type: none"> <li>1. Caso o não exista um grupo criado no GSN o sistema informa isso ao usuário.</li> <li>2. O usuário retorna a GSN para criar o evento/grupo.</li> <li>3. Executa novamente o passo 1 do [FP], agora com o evento criado/grupo.</li> </ol>
<b>Fluxo Secundário</b>	<ol style="list-style-type: none"> <li>1. Caso o sistema não consiga conexão com a rede social desejada</li> </ol>

<b>[FS02]</b>	o sistema informa o timeout ao usuário. 2. O usuário deve escolher se tenta novamente ou retorna ao menu.
<b>Fluxo Secundário [FS03]</b>	1. Caso o sistema não consiga criar a playlist na MSN o sistema deve informar a falha ao usuário. 2. O usuário deve escolher se tenta novamente ou retorna ao menu.

### 3.2.2. [UC02] Generate users Top10

<b>Identificador</b>	[UC02]
<b>Nome</b>	Generate users Top10
<b>Objetivo</b>	Criar uma lista dos 10 artistas mais tocados de um usuário
<b>Requisitos</b>	RF01; RF11; RF12; RF13
<b>Atores</b>	Facefy user, Music Social Network (MSN)
<b>Pré-condições</b>	O sistema deve ter uma lista de usernames de um evento para solicitar à MSN os dados de suas playlists.
<b>Frequência de uso</b>	Alta
<b>Pós-condições</b>	O sistema deve ter uma lista das músicas mais tocadas da lista de artistas dos usuários.
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O sistema requisita à MSN quais das pessoas da lista possuem conta. [FS01]</li> <li>2. A MSN responde com uma lista de usuários cadastrados.</li> <li>3. O sistema solicita à MSN os artistas das playlists dos usuários. [FS02]</li> <li>4. A MSN responde com os artistas dos usuários.</li> <li>5. O sistema organiza a lista de artistas e escolhe os mais citados e solicita à MSN as músicas mais tocadas desses artistas.</li> <li>6. A MSN responde com as músicas mais tocadas dos artistas</li> </ol>

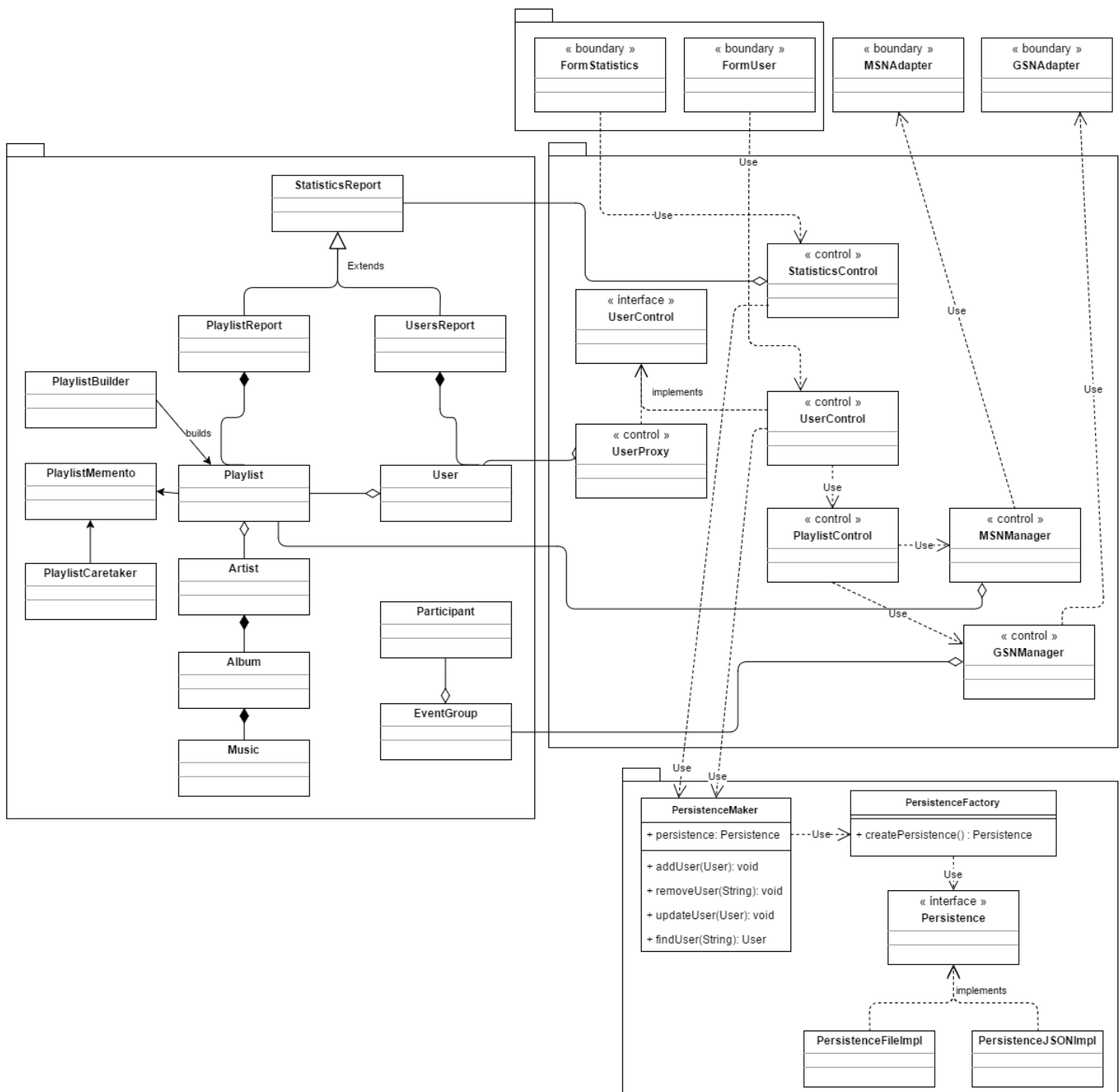
	pedidos. 7. O sistema gera uma lista com todas as músicas pedidas.
<b>Fluxo Secundário [FS01]</b>	1. Caso nenhum usuário da lista possua conta o sistema informa ao usuário e retorna ao menu inicial.
<b>Fluxo Secundário [FS02]</b>	1. Caso nenhum dos usuários listados tenha playlists cadastradas o sistema informa ao usuário e retorna ao menu inicial.

### 3.2.3. [UC03] Generate list of users

<b>Identificador</b>	[UC03]
<b>Nome</b>	Generate list of users
<b>Objetivo</b>	Criar uma lista dos usuários da Groups Social Network que confirmaram participação no evento ou fazem parte do grupo.
<b>Requisitos</b>	RF01; RF11; RF12;
<b>Atores</b>	Facefy user, Groups Social Network (GSN)
<b>Pré-condições</b>	O sistema deve ter conectado com sucesso à GSN.
<b>Frequência de uso</b>	Alta
<b>Pós-condições</b>	O sistema deve ter uma lista de usuários confirmados para o evento ou uma lista de usuários que fazem parte de um grupo.
<b>Fluxo Principal [FS]</b>	<ol style="list-style-type: none"> <li>1. O usuário escolhe se vai criar uma playlist baseada num grupo ou em um evento.</li> <li>2. O sistema se comunica com a GSN e mostra uma lista dos eventos/grupos que o usuário participa. [FS01]</li> <li>3. O usuário seleciona qual dos eventos/grupos ele quer utilizar para criação da playlist.</li> </ol>

	<ol style="list-style-type: none"> <li>4. O sistema solicita a GSN a lista de usuários confirmados no evento ou a lista de participantes do grupo. [FS02]</li> <li>5. A GSN responde com a lista de usuários confirmados no evento ou lista de participantes do grupo.</li> </ol>
<b>Fluxo Secundário [FS01]</b>	<ol style="list-style-type: none"> <li>1. Caso não participe de nenhum grupo ou não tenha algum evento cadastrado na GSN o sistema deve informar isso ao usuário.</li> <li>2. O usuário retorna à GSN para criar o evento ou grupo.</li> <li>3. O sistema retorna para o passo 1 do [FS]</li> </ol>
<b>Fluxo Secundário [FS02]</b>	<ol style="list-style-type: none"> <li>1. Caso o evento ou grupo selecionado não tenha participantes o sistema informa ao usuário.</li> <li>2. O usuário retorna à GSN para adicionar participantes ao evento ou grupo.</li> <li>3. O sistema retorna para o passo 1 do [FS]</li> </ol>

## 4. Diagrama de classes





## 5. Padrões de Design

Nesta seção vamos descrever os padrões de design que deverão ser utilizados na implementação do produto.

### 5.1. Template Method

O padrão Template Method será utilizado na criação dos relatórios. As classes envolvidas serão *StatisticsControl*, *StatisticsReport*, *PlaylistReport* e *UserReport*. A classe abstrata *StatisticsReport* vai ter o corpo padrão do relatório. As informações que serão geradas pelas classes concretas *PlaylistReport* e *UserReport*. A classe *StatisticsControl* vai fazer o controle dos relatórios gerados.

### 5.2. Factory

O padrão Factory será utilizado para a criação dos sistemas de persistência. A classe *PersistenceFactory* será a fábrica de *Persistence*. Essa fábrica será utilizada pela fachada para criar o modelo de persistência.

### 5.3. Adapter

O padrão Adapter será utilizado para comunicação do produto com as API's externas. A classe *MSNAdapter* vai fazer a tradução das chamadas da API da Music Social Network para gerar as entidades do sistema enquanto a classe *GSNAdapter* vai fazer a tradução da API da Groups Social Network para as entidades do nosso sistema.

### 5.4. Singleton

O padrão Singleton será utilizado na classe *UserControl* e *PlaylistControl* para garantir que apenas uma instância dessas classes serão criada e utilizada durante a execução do sistema.

### 5.5. Builder

O padrão builder será utilizado para a criação de uma playlist. A classe *PlaylistBuilder* será o builder da classe concreta *Playlist* - objeto mais complexo.

## **5.6. Prototype**

O padrão prototype será utilizado para a criação de uma gêneros musicais. Na aplicação a interface de prototipação é MusicPrototype.

## **5.7. Proxy**

O padrão proxy será utilizado para abstrair operações que seriam destinadas à UserControl. Na aplicação definimos UserProxy, classe que serve de fachada para operações destinadas a UserControl.

## **5.8. Memento**

O padrão memento será utilizado para salvar os estados durante a criação das playlists. Isso acontecerá nas classes PlaylistMemento e PlaylistCaretaker. Com isso a qualquer momento o Builder poderá salvar as alterações quando as faz, e poderá retornar a esses momentos anteriores específicos da criação.