

SIF1

Sistema de informação da Fórmula 1



Universidade do Minho
Escola de Engenharia

Miguel Esteves nº PG13544

Ricardo Santos nº PG12677

Tiago Correia nº PG13363

Índice

Introdução	3
Enunciado do problema	4
Análise do Problema	5
Funcionalidades	5
Personas	6
Infra-estrutura de uma corrida	8
Arquitectura	9
Arquitectura HA do sistema.....	9
Arquitectura da aplicação	11
Implementação.....	12
Decisões de Implementação	12
Modelação.....	14
Diagramas UML numa fase inicial	14
Diagramas UML	16
CTTEs.....	19
Interfaces.....	20
Conclusão	27

Introdução

A **Fórmula 1** é a mais popular modalidade de automobilismo do mundo no início do século XXI. Ela tem o objetivo de ser a categoria mais avançada do desporto a motor. É regulamentada pela *Fédération Internationale de l'Automobile* (FIA), com sede em Paris. O nome “Fórmula” refere-se a um conjunto de regras que os pilotos e os carros, representados pelos construtores, tem de obedecer. Um evento de Fórmula 1 está inserido num campeonato, que é realizado anualmente, e cada evento começa com um conjunto de treinos, em que cada piloto pode ou não efectuar, segue-se as provas de qualificação, que correspondem a eliminatórias em que se determina a grelha de partida, e finalmente a corrida.

Um carro de Fórmula 1 pode atingir os 360 km/h e por isso é necessário ter um sistema muito fiável e de alto desempenho para calcular os tempos dos pilotos em prova. Este sistema normalmente corresponde a um conjunto de transponders, que estão acoplados aos carros, e a um sistema de receptores de sinal que estão colocados em diversos pontos do circuito de modo a constituir uma série de sectores. Estes transponders transmitem sinais que são recepcionados pelos receptores ao longo da pista. Os sinais emitidos pelos transponders transmitem o identificador do transponder, bem como a qualidade dos sinais transmitidos de modo a poder ser calculado o tempo de cada piloto quando passa pelos diferentes pontos do circuito que tenham receptores. Estes sinais são depois passados a um aparelho designado “decoder” que descodifica estes sinais e permite obter as informações que necessárias ao cálculo do tempo do piloto. É aqui que entra o nosso sistema que será designado de “Sistema de Informação de Fórmula 1 (SIF1)”. Este recebe a informação do decoder e efectua as tarefas explicitadas no capítulo “enunciado do problema”.

O Sistema de informação da Fórmula 1 (SIF1) consiste numa aplicação web destinada à organização de corrida, que faz a gestão e marcação das corridas entre outras tarefas e o público que pode visitar as diversas informações sobre corridas, pilotos, equipas, entre outros.

No capítulo seguinte será descrito todas as funcionalidades do SIF1, seguido pelo capítulo da análise que explicitará ao pormenor as funcionalidades do sistema. De seguida, no capítulo das personas, será descrito as personas do sistema. No capítulo “Infra-Estrutura de uma corrida” será mostrado o funcionamento em conjunto dos transponders, receptores de sinal, decoder e SIF1. Seguem-se dois capítulos que descrevem a arquitectura do sistema. Finalmente, nos capítulos seguintes será explicado a implementação do SIF1.

Enunciado do problema

Actualmente na Fórmula 1, e em campeonatos semelhantes, a informação que é recolhida e disponibilizada aos utilizadores é muito vasta.

Além de se registar para cada prova a classificação final e a grelha de partida, com os respectivos tempos, o sistema de informação do campeonato regista as voltas de todos os pilotos (em qualificação, treinos e prova), bem assim como os tempos parciais (cada circuito é dividido em sectores) de cada volta.

Esta recolha de informação permite que seja possível determinar os melhores tempos de cada piloto por sector, determinar a volta ideal (com os melhores tempos por parcial independentemente do piloto), entre outra informação. Acrescem a estas consultas as que permitem saber a classificação actual dos pilotos e equipas.

No que concerne a aplicações cliente existe a página web onde depois de cada treino ou prova se apresenta a informação sintetizada e aplicações para as equipas e espectadores no circuito.

A aplicação para as equipas é constantemente actualizada e reflecte os tempos por volta e os parciais, permitindo comparar os tempos dos pilotos da equipa com os demais pilotos.

A aplicação dos espectadores permite analisar a informação piloto a piloto (com os seus tempos, parciais e pontos por prova), bem como aceder à informação de todos os pilotos.

As aplicações devem ainda fornecer informação estatística de diversa ordem como determinar o número médio de voltas por jogo de pneus (por paragem na box), velocidade máxima em cada prova, etc.

Análise do Problema

Funcionalidades

A informação disponibilizada neste sistema consiste essencialmente em:

- Classificação final de uma corrida
- Grelha de partida
- Tempos
- Voltas dos pilotos (em qualificação, treinos e prova)
- Tempos parciais
- Pesquisas, estatísticas e comparativos sobre pilotos, equipas, etc.

As operações de gestão correspondem resumidamente em:

- Criação, edição e remoção de campeonatos, circuitos, equipas e pilotos
- Administração de utilizadores

Os utilizadores que interagem com este sistema caracterizam-se pelas seguintes categorias:

- Cronometrista
- Gestor
- Juiz da partida
- Equipa (acesso a partir da Box)
- Administrador do sistema
- Utilizador do público privilegiado
- Utilizador comum

Personas

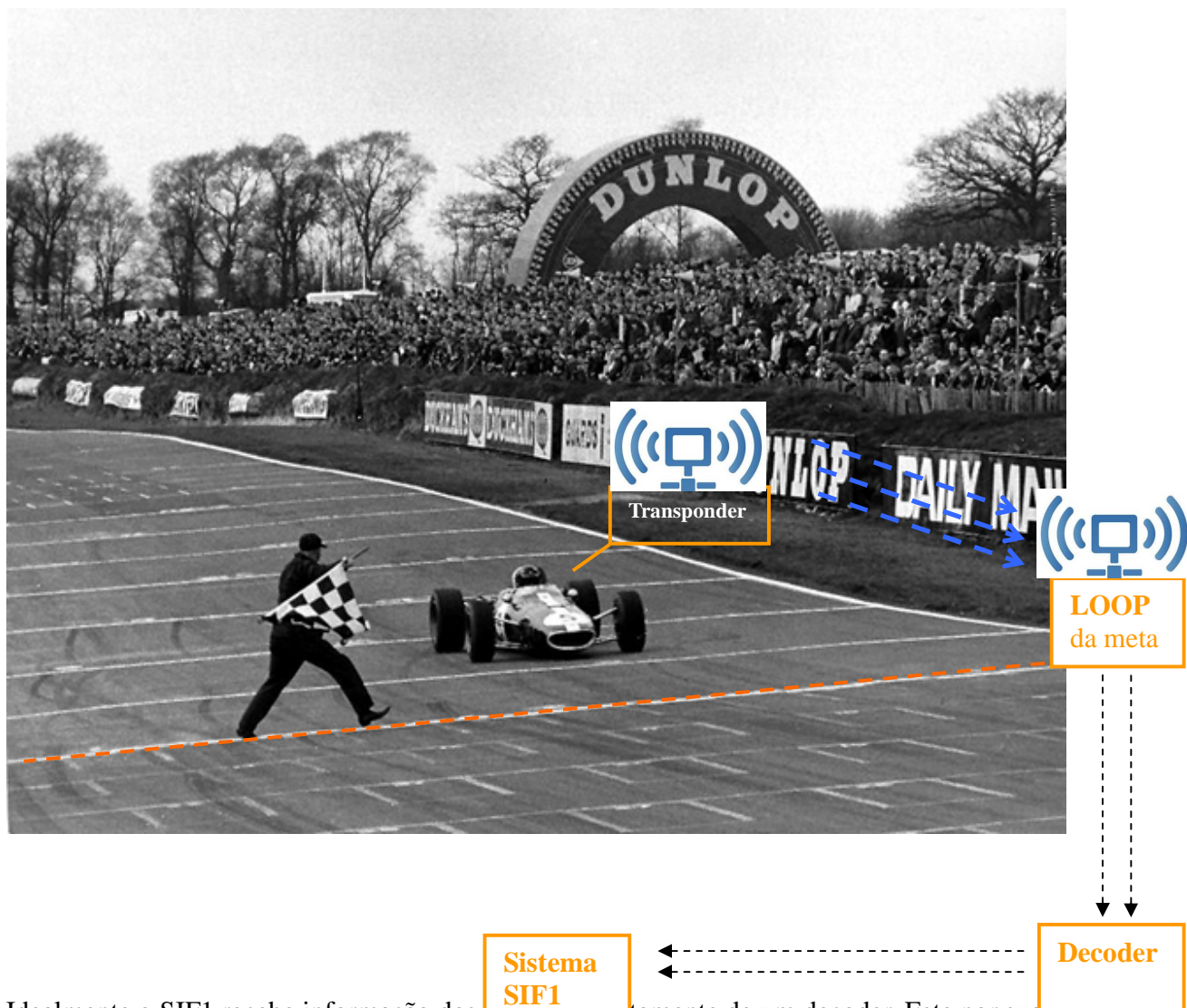
Segue-se uma série de perfis de utilizadores de acordo com as categorias de utilizadores do sistema.

Administrador do sistema	<p>Nome: Joaquim Silva</p> <p>Temperamento: Pacífico, pratica yoga.</p> <p>Nível de perícia: Utilizador intermédio. Sabe usar um sistema simples, preencher formulários e fazer login. Não precisa de feedback constante.</p> <p>Classe: Utilizador de suporte.</p> <p>Credencial: ADMIN</p> <p>Objectivo: Gerir utilizadores.</p> <p>Objectivo em detalhe: Ver todos os utilizadores, remover, editar informação pessoal e credencial de um utilizador e activar ou desactivar.</p>
Gestor	<p>Nome: Catarina Artur</p> <p>Temperamento: Impaciente, pratica kung fu.</p> <p>Nível de perícia: Utilizador intermédio. Sabe usar um sistema simples, preencher formulários e fazer login. Não precisa de feedback constante.</p> <p>Classe: Utilizador directo.</p> <p>Credencial: GESTOR</p> <p>Objectivo: Gerir campeonatos, circuitos, corridas, equipas e pilotos.</p> <p>Objectivo em detalhe: Ver listagem de campeonatos, criar, editar e removê-los, ver listagem de eventos de um campeonato, criar, editar ou removê-los, adicionar circuito, corrida, treino ou qualificação a um evento. Ver listagem de circuitos, criar, editar ou remover cricuítos, ver listagem de loops da um circuito, criar, editar ou remover loops. Ver listagem das equipas, criar, editar ou removê-las, inscrever equipa no campeonato actual, ver listagem de pilotos de uma equipa, adicionar ou remover pilotos da equipa. Ver listagem de pilotos, criar, editar ou remover.</p>
Cronometrista	<p>Nome: Alberto Carvalho</p> <p>Temperamento: Paciente, é pescador.</p> <p>Nível de perícia: Utilizador intermédio. Sabe usar um sistema simples, preencher formulários e fazer login. Não precisa de feedback constante.</p> <p>Classe: Utilizador directo.</p> <p>Credencial: CRONO</p> <p>Objectivo: Monitorizar cronometragem da corrida.</p> <p>Objectivo em detalhe: Dar início à sessão, dar início ao warmup, pausar ou parar a sessão, finalizar a sessão. Ver listagem de tempos por piloto por loop. Essa informação deve ter o número, nome do piloto, número do transponder, número de voltas, tempos da última volta, tempo total, hora actual, hits (nº de leituras que o loop fez ao transponder naquele instante), força de sinal de cada transponder e tempo de fotocélula. Também coloca manualmente o</p>

	tempo de fotocélula se o transponder falhar.
Juiz da Partida	<p>Nome: Jorge Fonseca</p> <p>Temperamento: Impulsivo, tem vários processos pendentes em tribunal.</p> <p>Nível de perícia: Utilizador intermédio. Sabe usar um sistema simples, preencher formulários e fazer login. Não precisa de feedback constante.</p> <p>Classe: Utilizador directo.</p> <p>Objectivo: Consultar Scoreboard em tempo real e atribuir penalizações aos pilotos.</p> <p>Objectivo em detalhe: A partir da scoreboard seleccionar o piloto que pretende penalizar. Ver os últimos registos de timing desse piloto com opção para invalidar volta e escrever comentários. Desclassificar piloto, aplicar castigo (alteração de tempo total, nº de voltas percorridas, pit stop ou drive by).</p>
Equipa (acesso a partir da box)	<p>Nome: Ana Diana</p> <p>Temperamento: Impaciente, querem ter feedback contante sobre informação da corrida pois investem muito dinheiro na fórmula 1.</p> <p>Nível de perícia: Utilizador inexperiente. Sabe usar fazer login.</p> <p>Classe: Utilizador remoto.</p> <p>Credencial: SBTR*</p> <p>Objectivo: Consultar Scoreboard em tempo real.</p> <p>Objectivo em detalhe: Ver listagem das posições dos pilotos na corrida, qualificação ou treino actual. A informação inclui posição, nome do piloto, número do piloto, gap (intervalo de tempo ao piloto que vai no primeiro lugar), intervalo (intervalo de tempo em relação ao piloto que vai atrás, tempo da última volta, tempo pelas diferentes secções, paragens nas boxes (pit), melhor tempo, tempo de qualificação. Deve ainda ter um placar com a melhor volta de um piloto, a volta ideal numa combinação das melhores secções e um painel com os castigos e outros acontecimentos da partida.</p>
Utilizador do público privilegiado	<p>Nome: Sandra Teresa</p> <p>Temperamento: Impaciente, pagou por isso acha que tem direito a informação em tempo real que nunca falhe.</p> <p>Nível de perícia: Utilizador inexperiente. Sabe usar fazer login.</p> <p>Classe: Utilizador remoto.</p> <p>Credencial: SBTR</p> <p>Objectivo: Consultar Scoreboard em tempo real.</p>
Utilizador comum	<p>Nome: Gustavo Costa</p> <p>Temperamento: Exigente. Gosta de fazer comparações entre sistemas web.</p> <p>Nível de perícia: Utilizador intermédio. Sabe usar um sistema simples, preencher formulários e fazer login. Não precisa de feedback constante.</p> <p>Classe: Utilizador remoto.</p> <p>Credencial: Nenhuma</p> <p>Objectivo: Consultar resultados finais.</p> <p>Objectivo em detalhe: Consultar resultados das equipas por campeonato. Consultar resultados dos pilotos por campeonato. Consultar resultados dos campeonatos por evento. Consultar resultados dos eventos/sessões por piloto por campeonato.</p>

Infra-estrutura de uma corrida

A seguinte figura mostra o funcionamento da infra-estrutura de uma corrida.



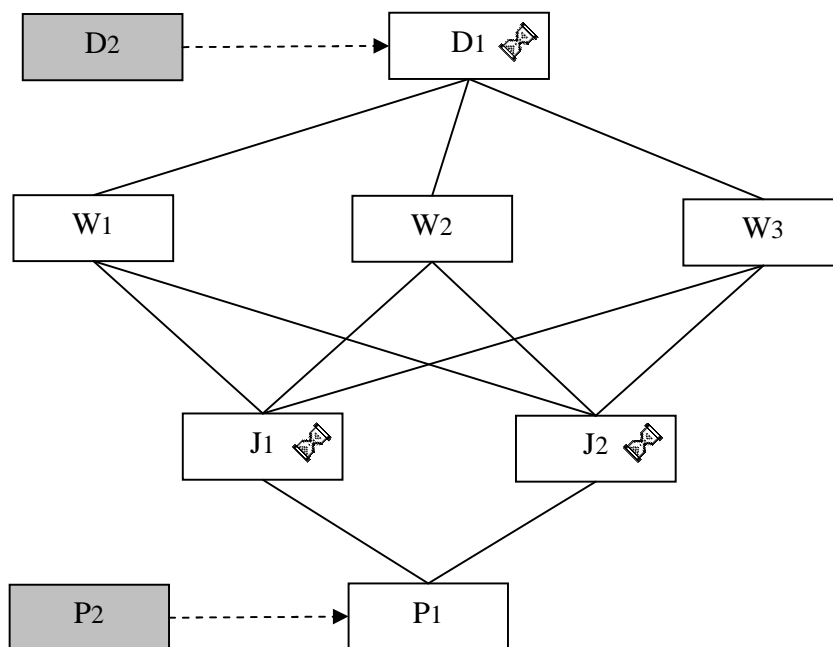
Idealmente o SIF1 recebe informação dos **Sistema SIF1** directamente de um decoder. Este por sua vez recebe directamente dos vários LOOPS. Neste caso apenas está exemplificado um. Um circuito divide-se em secções de acordo com quantos Loops tiver. Quando o carro passar na linha da meta, o transponder situado dentro deste vai comunicar com Loop transmitindo o seu identificador. O Loop encaminha para o decoder e este cria um registo com o id do carro mais o hora actual do mesmo.

Devido à falta de acesso aos protocolos do decoder esta parte será gerada virtualmente.

Arquitectura

Arquitectura HA do sistema

Considere-se o esquema HA da arquitectura low cost do SIF1



Director (D_x)

Heartbeat com D2 ↔ D1 e load balance para W₁, W₂, W₃.

Web Server (W_x)

Apache com o mod_jk para distribuir carga por J₁, J₂.

Application Server (Jx)

JBoss

Base de Dados (Px)

Base de dados MySQL/PostgreSQL com heartbeat entre P1 ↔ P2 e DRDB para replicação dos dados.



Locais onde é preciso configurar qual o timeout da sessão.

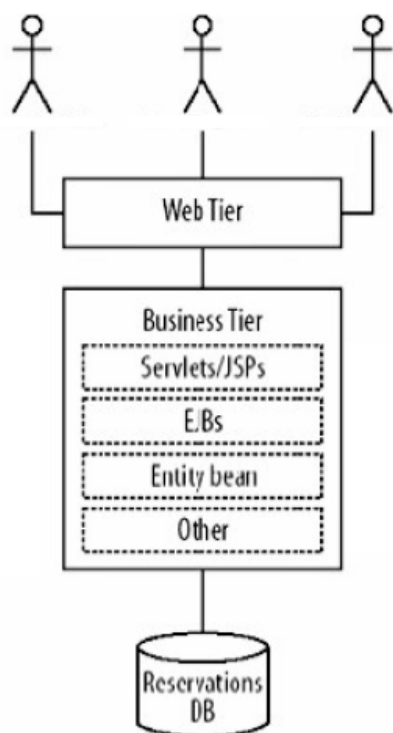
ipvs - persistência de sessão com base no IP origem + porta destino + timeout.

mod_jk - persistência de sessão com base no SESSIONID.

jboss - persistência de sessão com base no timeout e no pedido deliberado para a sua destruição.

Arquitectura da aplicação

O SIF1 é uma aplicação construída em J2EE a correr em JBoss. O modelo de desenvolvimento é n-tier como se pode ver na figura seguinte.

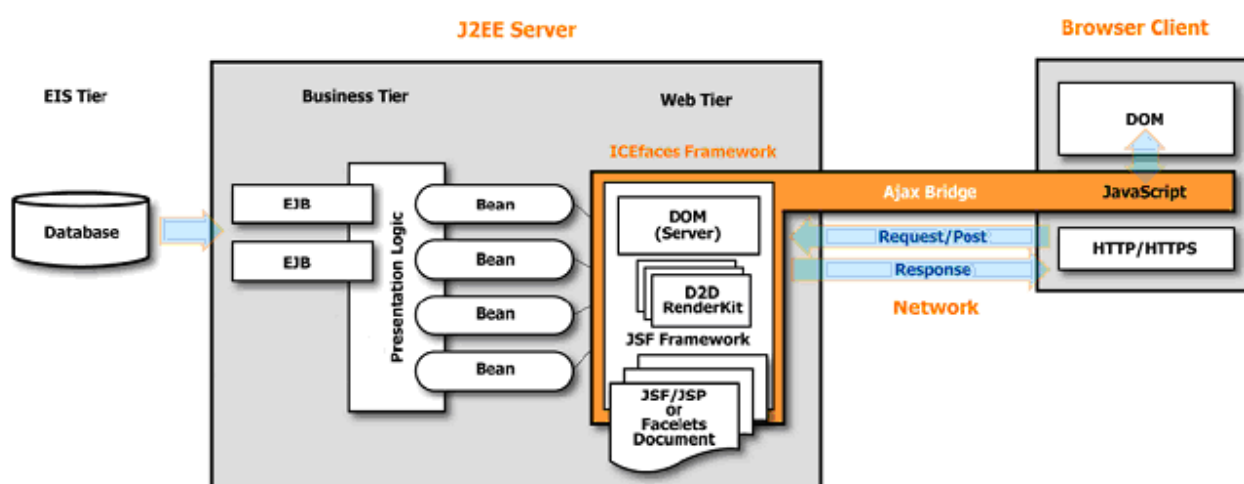


Usou-se EJBs para a camada de negócio e Entity beans para as entidades dos dados. Para ligar à base de dados usou-se Hibernate.

Para a camada da apresentação usou-se servlets JSF. Usou-se a framework ICEfaces para desenvolvimento pois é feito com base em requests assíncronos (AJAX), suporta Ajax Push (Comet) e possibilita um maior controlo e segurança sobre o cliente devolvendo-lhe uma maior experiência em usabilidade. O Ajax Push foi o principal factor para o uso desta framework pois este consiste em haver uma conexão permanente cliente/servidor onde o cliente não precisa de fazer requests para receber actualizações na página. Tal acontece pois a página é renderizada no servidor. A DOM tree do cliente será actualizada apenas no conteúdo alterado no servidor.

Isto é importante para o projecto pois em interfaces como o scoreboard em tempo real deve mostrar ao cliente os tempos a decorrer naquele preciso momento sem obrigar ao utilizador fazer refresh à página ou mesmo à sessão.

Segue-se um esquema elucidativo do funcionamento do ICEfaces.



Implementação

Decisões de Implementação

Nos dias que correm as aplicações web estão a ganhar terreno às aplicações locais, devido à grande expansão do mercado da internet. Por isso desenvolvemos uma aplicação web que além de apresentar informação estática, é possível realizar todas as tarefas de gestão de corridas de Fórmula 1. Para desenvolver a aplicação web tivemos que tomar algumas decisões de ordem arquitectural.

A aplicação teria que estar ligada com uma Base de Dados, de forma a poder guardar todos os dados necessários de modo persistente. Ao longo da execução de uma aplicação, a persistência de dados é factor crítico, pois determina o modelo de acesso aos dados, assim como o desempenho, a facilidade de acesso, a dificuldade de manutenção, entre outros. A nossa aplicação não é excepção, logo era necessário decidirmos a melhor forma de tornar os dados persistentes, tendo em conta que estávamos a utilizar java. Já que excluímos ao princípio a utilização de Base de Dados OO, devido à falta de confiabilidade que estas ainda apresentam, e uma vez que a utilização de JDBC implicaria que transformasse-mos manualmente o nosso modelo OO num modelo relacional, decidimos utilizar um sistema ORM (Object-relational mapping). Esta ferramenta mapea as entidades descritas no modelo OO em tabelas na base de dados relacional sem que o programador necessite de conhecer o modelo relacional da aplicação. Deste modo, e de entre as inúmeras ferramentas ORM, escolhemos a que mais se adequa à linguagem JAVA que é a framework Hibernate. O Hibernate é uma Framework open-source que permite mapear a representação dos dados de um modelo de objectos para um modelo de dados relacional e seu respectivo esquema na Base de Dados. A configuração do mapeamento entre as classes pode ser feita através do XML ou usando anotações no código das classes (*Hibernate Annotations*). E preferimos utilizar as anotações no código uma vez que é mais lógico e sabe-se à partida no código JAVA quais são as entidades e o que é que elas representam na Base de Dados.

Como a aplicação tem que depender de tecnologia e como esta está em constante evolução decidimos criar, em primeiro lugar, um modelo independente de tecnologia (PIM) e de seguida um modelo dependente de tecnologia (PSM). Desta forma será mais fácil mudar a tecnologia utilizada na aplicação, basta utilizar toda a lógica do modelo independente de tecnologia e criar outro modelo dependente de tecnologia. Na nossa solução criamos um PIM e um PSM para a camada de negócio e para a camada de dados. Desta forma fazer a manutenção da nossa aplicação e mudar, por exemplo, a tecnologia utilizada na camada de dados, é muito mais simples, não é necessário apagar código, basta criar outra camada de dados (PSM) com a tecnologia desejada.

Para desenvolver a camada de negócio utilizou-se os EJB (*Enterprise Java Beans*) que fornecem um rápido e simplificado desenvolvimento de aplicações Java baseado em componentes distribuídas, transacionais, seguras e portáteis. Através do EJB utiliza-se a infraestrutura do servidor de aplicacional voltada para o desenvolvimento de aplicações de missão crítica (de alta importância para a empresa) e de aplicações empresariais em geral.

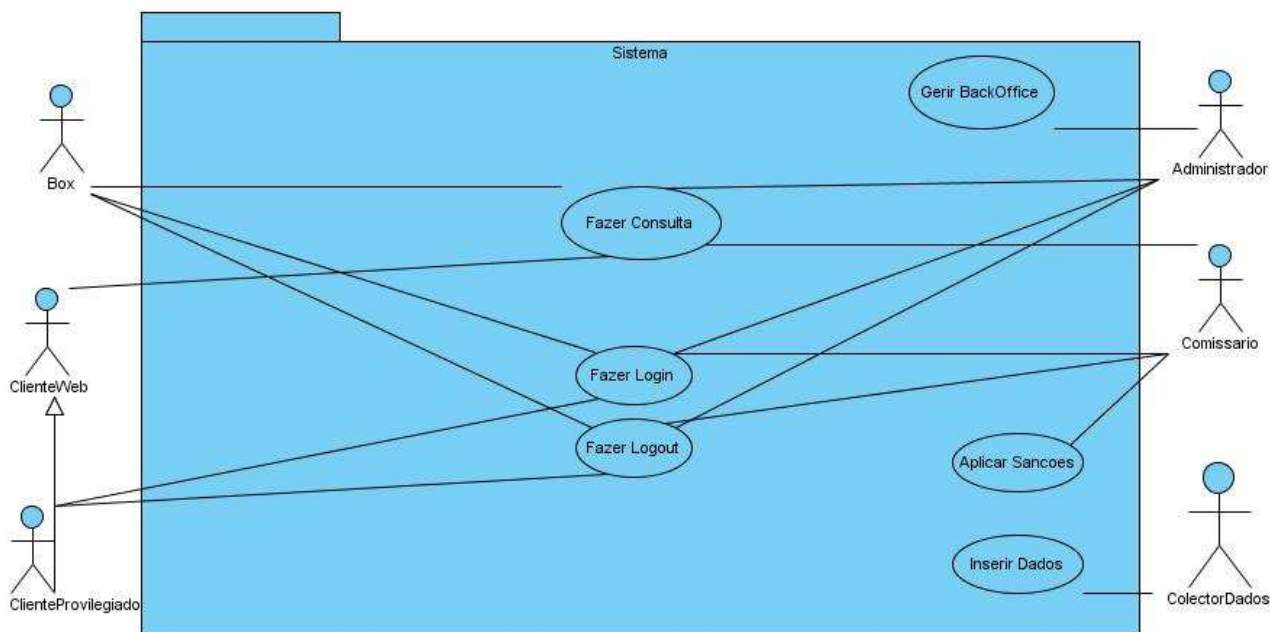
No que toca à camada de apresentação era necessário escolher qual o tipo de tecnologia a utilizar. A escolha da tecnologia a utilizar é influenciada desde logo pelo tipo de disponibilidade que queremos dar ao nosso sistema, na medida em que, se queremos um sistema que irá funcionar na máquina local da pessoa ou se por outro lado é um sistema acedido remotamente através de um browser. No nosso caso específico, vamos ter um sistema que vai ser acedido pelos utilizadores através de um browser, logo restringe desde logo as tecnologias a utilizar. Como todo o sistema foi desenvolvido em Java, optamos por utilizar JSP para a nossa camada de apresentação, uma vez que iria facilitar em muito o desenvolvimento do sistema. Mas para facilitar a escrita do JSP utilizou-se a a framework IceFaces que é parecida ao JSF. As razões que nos levaram a escolher esta framework são as seguintes:

- É open-source
- Componente de Ajax é muito fácil de desenvolver
- Compatibilidade com um grande número de servidores aplicacionais
- Utiliza Ajax Push
- Segurança
- É standard uma vez que é uma solução JavaEE

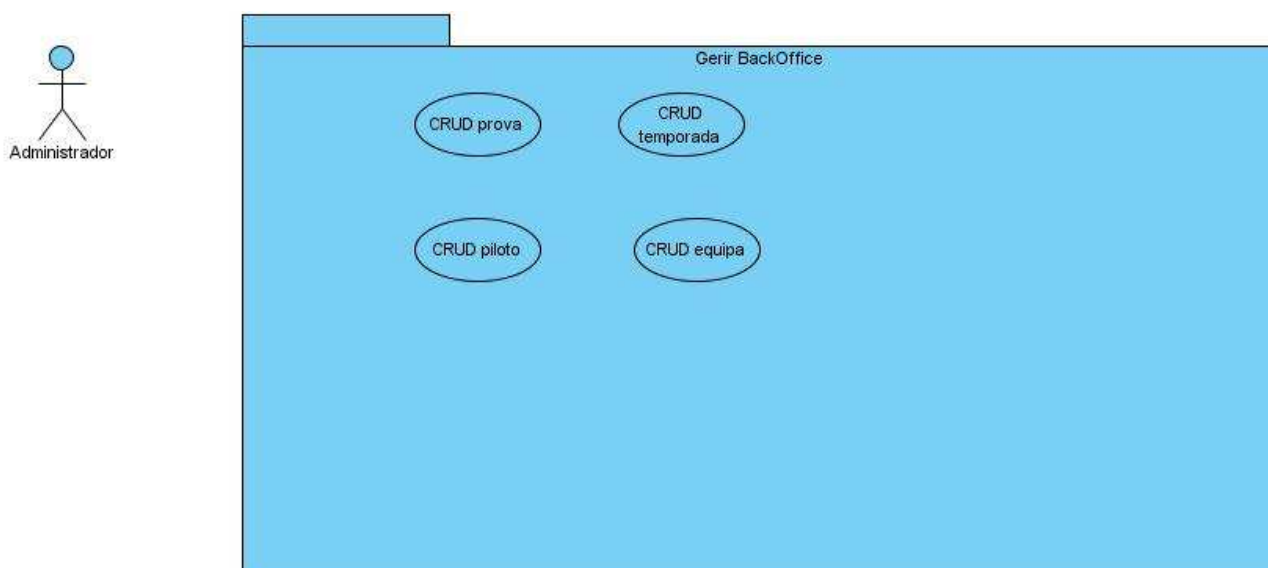
Modelação

Diagramas UML numa fase inicial

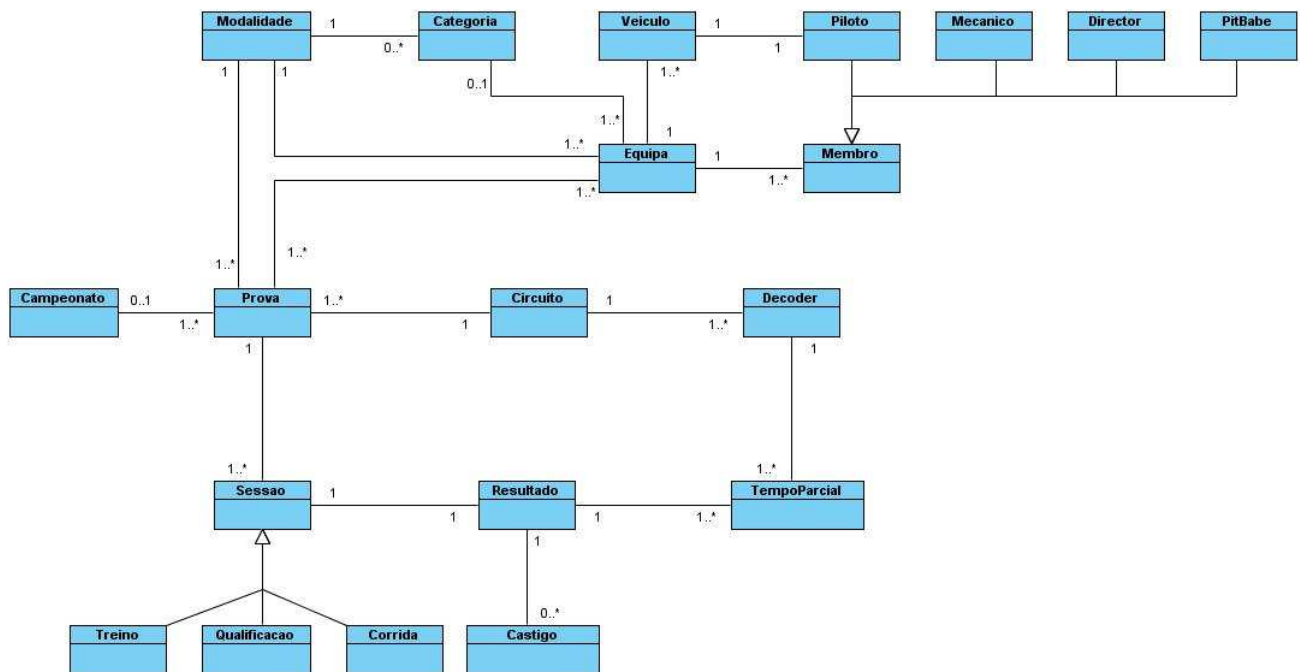
Use cases do sistema:



Use cases da gestão do back-office:



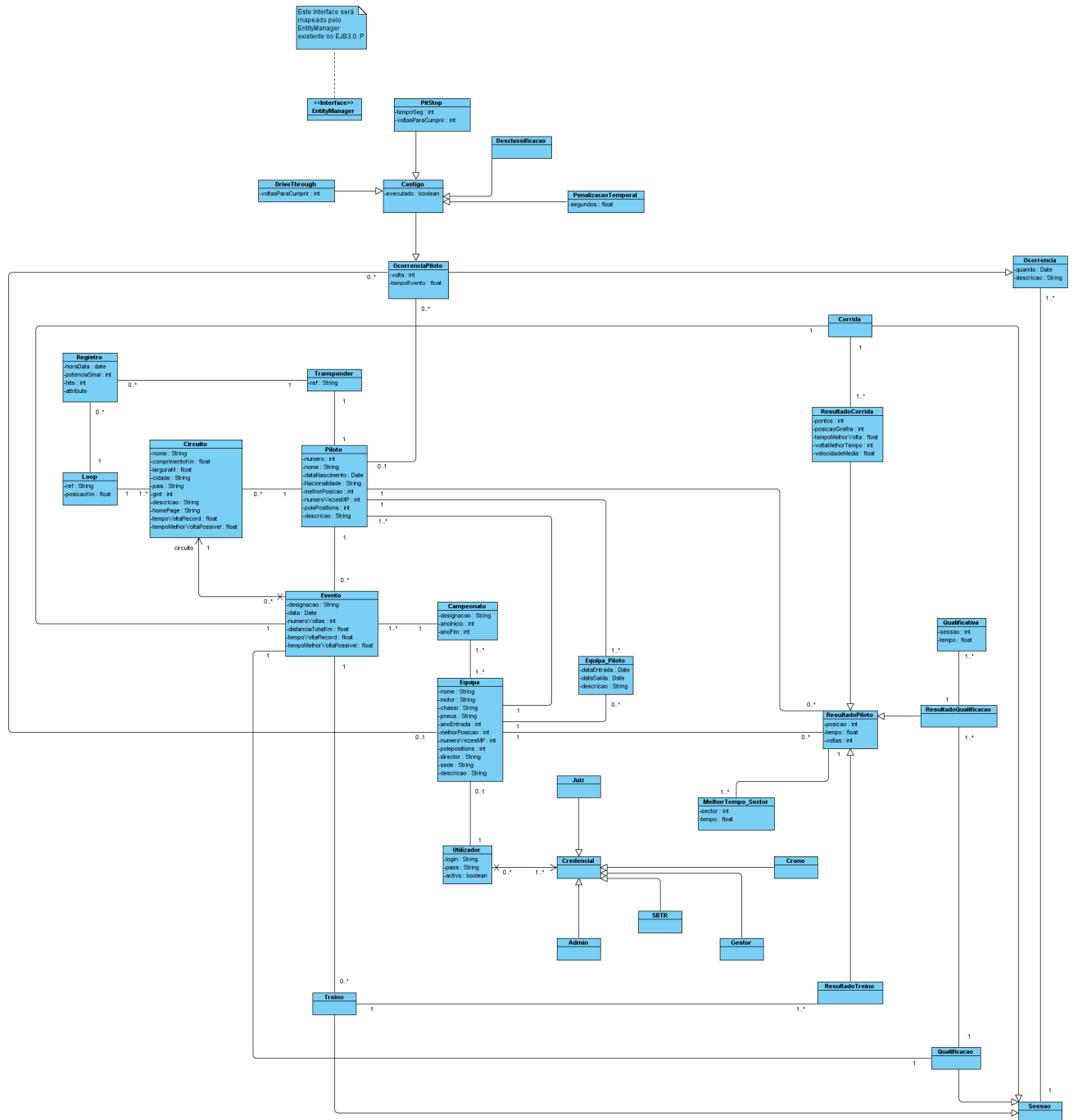
Diagramas de classe:



Diagramas UML

Diagramas PIM

Data-tier:



Business-tier:

GerirUtilizador
-utilizadorSel : Utilizador
+guardarAlteracoes() : void
+adicionarCredencial(credencial : Credencial) : void
+removerCredencial(credencial : Credencial) : void
+getListaUtilizadores() : Utilizador [0..*]
+removerUtilizador() : void

IniciarSessao
-utilizador : Utilizador
+login(user : String, pass : String) : boolean
+logout() : void
+isLoggedIn() : boolean

Registar
-utilizador : Utilizador
+criarNovoUtilizador() : void
+submeter() : void

GerirCampeonato
-campeonatoSel : Campeonato
-eventoSel : Evento
-circuitoSel : Circuito
+getListaCampeonatos() : Campeonato [0..*]
+getListaEventos() : Evento [0..*]
+getListaCircuitos() : Circuito [0..*]
+criarNovoCampeonato() : void
+criarNovoEvento() : void
+guardarAlteracoes() : void
+removerCampeonatoSel() : void
+removerEventoSel() : void

GerirCircuito
-circuitoSel : Circuito
-loopSel : Loop
+guardarAlteracoes() : void
+getListaCircuitos() : Circuito [0..*]
+getListaLoops() : Loop [0..*]
+criarNovoLoop() : void
+criarNovoCircuito() : void
+removerLoopSel() : void
+removerCircuitoSel() : void

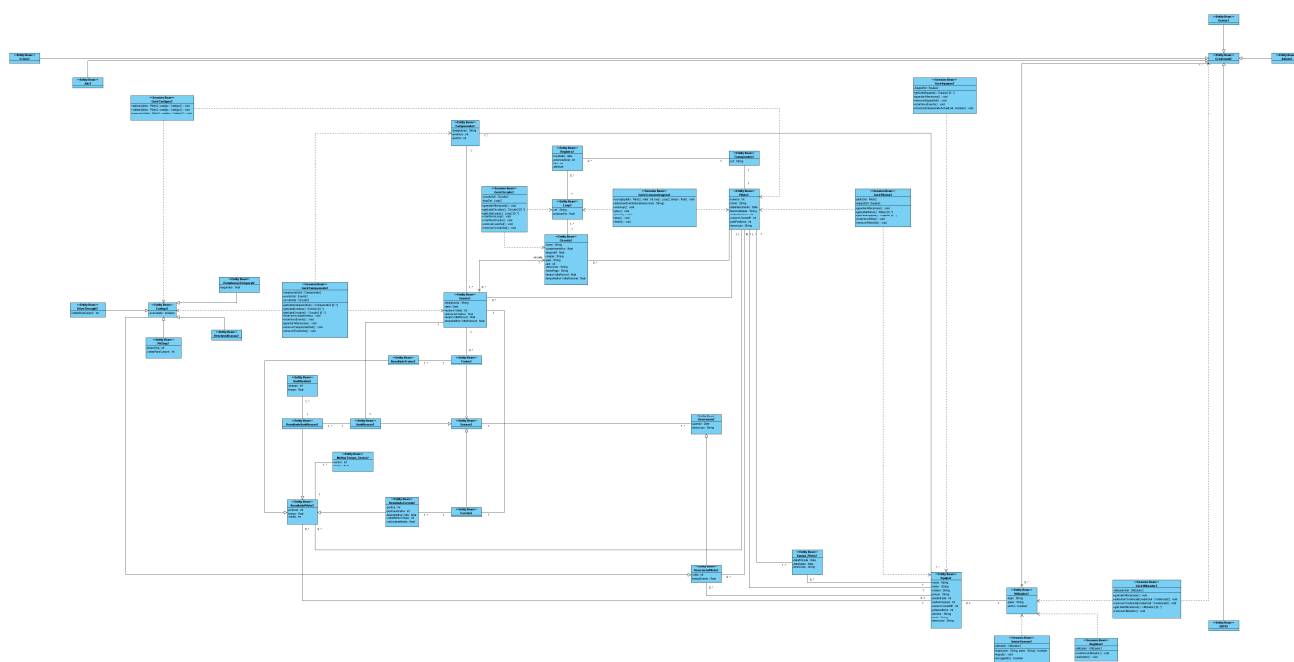
GerirCronometragem
+corrigir(piloto : Piloto, volta : int, loop : Loop, tempo : float) : void
+adicionarEventoGeral(descricao : String)
+warmup() : void
+play() : void
+pause() : void
+stop() : void
+finish() : void

GerirEquipas
-equipaSel : Equipa
+getListaEquipas() : Equipas [0..*]
+guardarAlteracoes() : void
+removerEquipaSel() : void
+criarNovoEvento() : void
+checkedCampeonatoActual(val : boolean) : void

GerirPilotos
-pilotoSel : Piloto
-equipaSel : Equipa
+guardarAlteracoes() : void
+getListaPilotos() : Piloto [0..*]
+getListaEquipas() : Equipa [0..*]
+criarNovoPiloto() : void
+removerPilotoSel() : void

GerirCastigos
+aplicar(piloto : Piloto, castigo : Castigo) : void
+validar(piloto : Piloto, castigo : Castigo) : void
+remover(piloto : Piloto, castigo : Castigo) : void

Diagrama PSM:



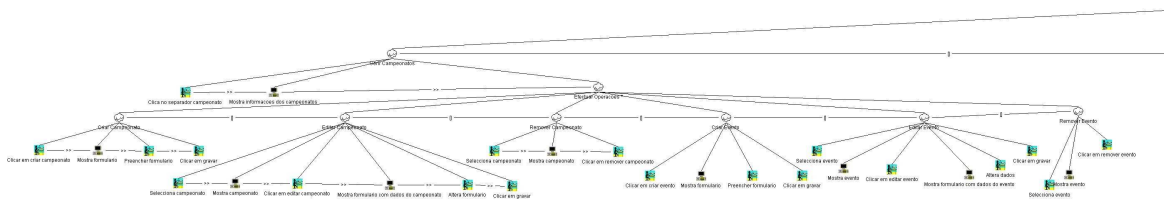
CTTEs

Gestor:

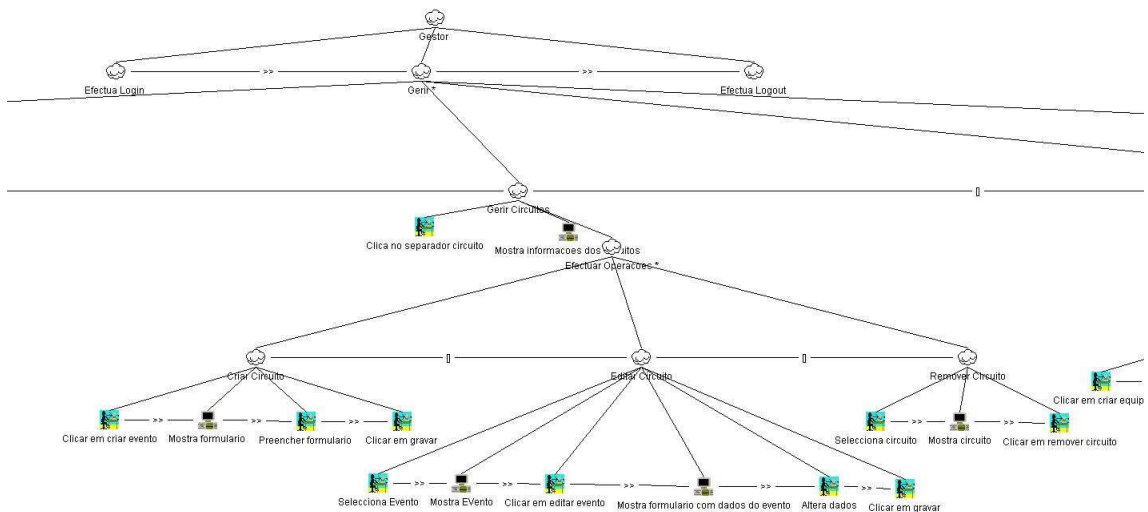
Vista geral



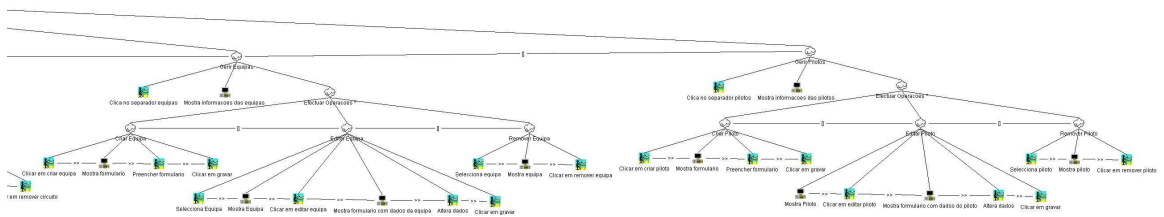
Parte esquerda:



Parte central:



Parte direita:




Interfaces

Na construção das interfaces tomamos em conta o conhecimento adquirido durante a UC de sistemas interactivos. No desenvolvimento de aplicações que tenha uma componente interactiva muito grande é necessário seguir algumas regras que permitem criar interfaces apropriadas aos utilizadores que irão usar a aplicação. Estas regras correspondem a *patterns* de desenvolvimento de interfaces interactivas. Estes *patterns* estão disponíveis no livro “*Designing Interfaces: Patterns for Effective Interaction Design*”. Deste modo, utilizamos alguns desses *patterns* tais como:

- Wizard
- Clear Entry Points
- Global Navigation
- Card Stack
- Responsive Enabling
- Row Striping
- Sortable Table
- Fill-in-the-Blanks
- Dropdown Chooser

Login:




Login

[Registar](#)
[Login](#)
[Inicio](#)
[Galeria](#)
[Resultados](#)

username:

password:

Registar:




Registar

[Registar](#)
[Login](#)
[Inicio](#)
[Galeria](#)
[Resultados](#)

Nome:

Email:

Data de nascimento: 

Username:


Password:

Confirmar password:

Credenciais:

- ☐ Juiz
- ☐ SBTR
- ☐ Gestor
- ☐ Crono
- ☐ Admin

Admin:



Administração

[Registar](#)
[Login](#)
[Início](#)
[Galeria](#)
[Resultados](#)

username	nome	datanasc	estado
vasco	Vasco Oliveira	06/06/1986	offline
tiago	Tiago Correia	13/10/1986	offline
rui	Rui Veloso	20/04/1985	offline
ricardo	Ricardo Santos	14/05/1984	offline
nuno	Nuno Correia	06/03/1985	offline
miguel	Miguel Esteves	08/08/1985	offline
str	John Doe STR	20/02/1980	offline
force	John Doe Force	20/02/1980	offline
bmw	John Doe BMW	20/02/1980	offline
guest9	John Doe	06/07/1969	offline
guest8	John Doe	06/07/1968	offline
guest7	John Doe	06/07/1967	offline
guest6	John Doe	06/07/1966	offline
guest5	John Doe	06/07/1965	offline
guest49	John Doe	06/07/2009	offline

Nome:

Email:

Data de nascimento:

Username:


Credenciais:

☐ Gestor
☐ Admin
☐ Juiz
☐ Crono
☐ SBTR

Equipa:

☐ activo

Gestor (Campeonatos):



Gestor

[Registar](#)
[Login](#)
[Início](#)
[Galeria](#)
[Resultados](#)

Campeonato
Criar Campeonato
Editar Campeonato
Remover Campeonato

Adicionar Evento
Editar Evento
Remover Evento

Mostrar Treinos
Adicionar Treino
Editar Treino
Remover Treino

Circuito

Equipas

Pilotos

Designação:

Ano de início:

Designação:

Data:

Número de voltas:

Distância total (km):

Circuito:

Corrida (data):

Qualificação (data):


Treino

Data

Projecto Integrado
Mestrado em Informática - Engenharia de Aplicações

22

Gestor (Circuitos):



Gestor

[Registrar](#)
[Login](#)
[Inicio](#)
[Galeria](#)
[Resultados](#)


[Campeonato](#) **Circuito** [Equipas](#) [Pilotos](#)

[Criar Circuito](#)
[Editar Circuito](#)
[Remover Circuito](#)

Nome:
País:
Cidade:
Comprimento (km):
Largura (m):
GMT:
Homepage:
Descrição:
Imagem: [Procurar...](#) [Upload](#)

[Mostrar Loops](#)
[Adicionar Loop](#)
[Editar Loop](#)
[Remover Loop](#)

Gestor (Equipas):



Gestor

[Registrar](#)
[Login](#)
[Inicio](#)
[Galeria](#)
[Resultados](#)


[Campeonato](#) [Circuito](#) **Equipas** [Pilotos](#)

[Criar Equipa](#)
[Editar Equipa](#)
[Remover Equipa](#)

Nome:
Motor:
Chassi:
Pneus:
Director:
Sede:
Descrição:
Inscrito em: ☐
Imagem: [Procurar...](#) [Upload](#)

[Pilotos](#)

Gestor (Pilotos):



Gestor

[Registrar](#)
[Login](#)
[Inicio](#)
[Galeria](#)
[Resultados](#)

CampeonatoCircuitoEquipas**Pilotos**

Criar Piloto

Editar Piloto

Remover Piloto

Nome:

Data de nascimento:

Nacionalidade:

Transponder:

Descrição:

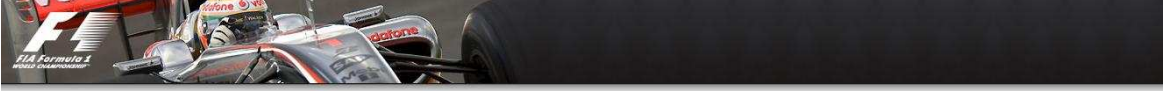
Equipa:

Imagem:

Procurar...

Upload

Cronometrista:



Crono

[Registrar](#)
[Login](#)
[Inicio](#)
[Galeria](#)
[Resultados](#)

campeonato10evento7_c10Corrida

start

stop

avancar para Q2

gravar

#	Nº	Piloto	Equipa	Transponder	Sector	Loop	Tempo do Sector	Tempo total	Tempo de Medição	Sinal	Hits	Volta	Pit
30	16	Kazuki Nakajima	Williams-Toyota	t16	2	m_s3	00:33,203	00:01:04,859	30/07/2009 10:02:43,671	31	75	1	false
29	5	Sebastien Bourdais	STR-Ferrari	t5	2	m_s3	00:24,172	00:01:01,344	30/07/2009 10:02:40,156	82	29	1	false
28	11	Mark Webber	RBR-Renault	t11	2	m_s3	00:20,343	00:00:59,734	30/07/2009 10:02:38,546	47	74	1	false
27	9	Felipe Massa	Ferrari	t9	2	m_s3	00:26,438	00:00:59,344	30/07/2009 10:02:38,156	91	22	1	false
26	6	Adrian Sutil	Force India-Mercedes	t6	2	m_s3	00:26,453	00:00:59,281	30/07/2009 10:02:38,093	89	53	1	false
25	10	Nico Rosberg	Williams-Toyota	t10	2	m_s3	00:27,875	00:00:59,094	30/07/2009 10:02:37,906	49	86	1	false
24	2	Rubens Barrichello	Brawn-Mercedes	t2	2	m_s3	00:35,969	00:00:58,438	30/07/2009 10:02:37,250	88	55	1	false
23	12	Jarno Trulli	Toyota	t12	2	m_s3	00:26,171	00:00:56,609	30/07/2009 10:02:35,421	85	22	1	false
22	17	Sebastian Vettel	RBR-Renault	t17	2	m_s3	00:29,000	00:00:54,719	30/07/2009 10:02:33,531	33	48	1	false
21	18	Heikki Kovalainen	McLaren-Mercedes	t18	2	m_s3	00:20,234	00:00:50,828	30/07/2009 10:02:29,640	82	53	1	false
20	3	Lewis Hamilton	McLaren-Mercedes	t3	2	m_s3	00:22,469	00:00:49,766	30/07/2009 10:02:28,578	67	18	1	false
19	11	Mark Webber	RBR-Renault	t11	1	m_s1	00:36,032	00:00:39,391	30/07/2009 10:02:18,203	94	72	1	false
18	8	Kimi Raikkonen	Ferrari	t8	1	m_s1	00:35,641	00:00:38,688	30/07/2009 10:02:17,500	95	46	1	false
17	14	Nick Heidfeld	BMW Sauber	t14	1	m_s1	00:35,297	00:00:38,453	30/07/2009 10:02:17,265	64	25	1	false
16	13	Timo Glock	Toyota	t13	1	m_s1	00:34,625	00:00:37,844	30/07/2009 10:02:16,656	87	8	1	false
15	5	Sebastien Bourdais	STR-Ferrari	t5	1	m_s1	00:34,281	00:00:37,172	30/07/2009 10:02:15,984	35	40	1	false
14	19	Nelsinho Piquet	Renault	t19	1	m_s1	00:33,344	00:00:36,844	30/07/2009 10:02:15,656	44	86	1	false

SBTR (Utilizador privilegiado/Equipa na Box)

Scoreboard

[Registar](#)
[Login](#)
[Inicio](#)
[Galeria](#)
[Resultados](#)

Pos	Nº	Equipa	Piloto	GAP	Intervalo	S1	S2	S3	S4	Tempo	PIT
1	15	BMW Sauber	Robert Kubica	00:05:23,422	volta: 4	00:32,469	00:21,750	00:20,812	00:24,766	01:45,109	1
2	6	Force India-Mercedes	Adrian Sutil	00:02,125	00:02,125	00:20,063	00:37,484	00:28,391	00:33,265	02:10,265	0
3	12	Toyota	Jarno Trulli	00:07,328	00:05,203	00:31,219	00:32,578	00:27,766	00:35,234	02:04,859	0
4	17	RBR-Renault	Sebastian Vettel	00:11,593	00:04,265	00:33,219	00:20,125	00:31,125	00:27,390	01:44,656	0
5	16	Williams-Toyota	Kazuki Nakajima	00:15,859	00:04,266	00:32,578	00:27,125	00:22,297	00:28,781	01:40,265	0
6	4	Renault	Fernando Alonso	00:16,625	00:00,766	00:33,891	00:28,969	00:32,750	00:27,625	01:58,109	0
7	7	Force India-Mercedes	Giancarlo Fisichella	00:19,063	00:02,438	00:36,844	00:29,094	00:22,187	00:26,813	01:54,938	3
8	8	Ferrari	Kimi Raikkonen	00:19,860	00:00,797	00:20,891	00:26,000	00:23,203	00:30,094	01:40,188	0
9	2	Brawn-Mercedes	Rubens Barrichello	00:31,421	00:11,561	00:22,781	00:24,188	00:25,516	00:31,968	01:44,453	2
10	20	STR-Ferrari	Sebastien Buemi	00:32,750	00:01,329	00:33,860	00:38,984	00:21,984	00:25,516	02:00,344	0
11	19	Renault	Nelsinho Piquet	00:36,953	00:04,203	00:33,734	00:23,078	00:23,453	00:36,797	01:57,062	0
12	18	McLaren-Mercedes	Heikki Kovalainen	00:42,828	00:05,875	00:27,953	00:23,125	00:28,313	00:20,187	01:39,578	0
13	11	RBR-Renault	Mark Webber	00:43,593	00:00,765	00:35,015	00:24,797	00:30,922	00:35,078	02:05,812	1
14	3	McLaren-Mercedes	Lewis Hamilton	00:45,390	00:01,797	00:25,734	00:31,766	00:24,938	00:31,015	01:53,453	1
15	13	Toyota	Timo Glock	00:46,781	00:01,391	00:36,360	00:32,672	00:32,812	00:30,547	02:12,391	0
16	9	Ferrari	Felipe Massa	00:49,906	00:03,125	00:30,906	00:26,063	00:21,937	00:37,453	01:56,359	0
17	14	BMW Sauber	Nick Heidfeld	00:50,734	00:00,828	00:29,937	00:22,922	00:38,172	00:31,344	02:02,375	0
18	5	STR-Ferrari	Sebastien Bourdais	1 VA	1 VA	00:24,922	00:35,031	00:29,609	00:37,797	02:15,203	0
19	10	Williams-Toyota	Nico Rosberg	1 VA	00:02,031	00:35,953	00:28,203	00:32,953	00:36,547	02:19,297	0
20	1	Brawn-Mercedes	Jenson Button	1 VA	00:16,437	00:37,453	00:37,532	00:37,281	00:39,734	02:12,797	0

Utilizador comum (Consultas de resultados):



Resultados

[Registar](#)
[Login](#)
[Inicio](#)
[Galeria](#)
[Resultados](#)

Campeonato							
		Equipa	Piloto				
campeonato26	evento7_c26	Corrida					
Posição	Número	Piloto	Equipa	Volts	Tempo	Grelha	Pontos
1	15	Robert Kubica	BMW Sauber	60	00:44:06,918	1	10
2	14	Nick Heidfeld	BMW Sauber	60	00:45:26,348	2	8
3	1	Jenson Button	Brawn-Mercedes	60	00:46:34,509	4	6
4	2	Rubens Barrichello	Brawn-Mercedes	60	00:48:49,776	3	5
5	8	Kimi Raikkonen	Ferrari	60	00:50:51,243	6	4
6	7	Giancarlo Fisichella	Force India-Mercedes	60	00:52:20,115	8	3
7	9	Felipe Massa	Ferrari	60	00:52:23,579	5	2
8	4	Fernando Alonso	Renault	60	00:53:34,649	14	1
9	6	Adrian Sutil	Force India-Mercedes	60	00:54:39,279	7	0
10	17	Sebastian Vettel	RBR-Renault	60	00:55:26,175	12	0

Galeria:



Galeria



Registrar
Login
Inicio
Galeria
Resultados

Circuitos

Equipas

Pilotos

Nome da equipa
BMW Sauber
Brawn-Mercedes
Ferrari
Force India-Mercedes
McLaren-Mercedes
RBR-Renault
Renault
STR-Ferrari
Toyota
Williams-Toyota
<< 1-10 de 10 >>

Nome: Ferrari

Motor: motor4

Chassi: chassi4

Pneus: pneus4

Ano de Entrada: 2004

Melhor Posicao: 0

Número de vezes melhor posição: 0

Polepositions: 0

Director: director4

Sede:

Descrição:

Pilotos: Felipe Massa
Kimi Raikonen

Conclusão

Com a elaboração deste projecto ficamos mais cientes do processo de desenvolvimento de software e de toda a arquitectura que é necessária para suportar tal software. Ficamos a conhecer novas tecnologias e metodologias de desenvolvimento. Todo o funcionamento de um sistema de informação de Formula 1 também ficou conhecido.

Concluindo, com uma breve opinião pessoal, a evolução é sempre uma motivação, ainda mais quando é despertada tanto a nível de desenvolvimento como pós-conclusão do trabalho. O nosso objectivo é aprender a desenvolver aplicações web e interiorizar os conceitos que temos vindo a falar. A nossa motivação no fundo é saber que estamos a desenvolver algo de útil para o mercado de trabalho.