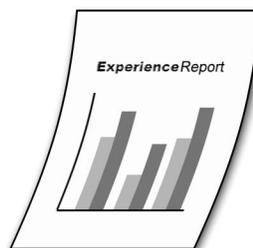


iSeries



# Descrição geral do cenário de PME da iDrink e de Cola Connections

## Relatório da **Experiência**





iSeries



# Descrição geral do cenário de PME da iDrink e de Cola Connections



---

# Índice

## Capítulo 1. Descrição geral do cenário de PME da iDrink e de Cola Connections 1

Descrição geral da iDrink . . . . .	1
Descrição geral de Cola Connections . . . . .	2
Descrição geral do ambiente . . . . .	3
Descobertas de PME. . . . .	4

## Capítulo 2. Ambiente do WebSphere Application Server - Express. . . . . 7

Descrição geral do WebSphere <sup>(R)</sup> Application Server	7
Fluxo da aplicação . . . . .	8
Ambiente de programação. . . . .	9
Detalhes da aplicação . . . . .	9
Pontos de concepção de aplicações . . . . .	10
UserServlet . . . . .	14

CustomerServlet. . . . .	15
EmployeeServlet. . . . .	18
SupplierServlet . . . . .	27
Instalação da aplicação para empresas . . . . .	30
Observações do WebSphere . . . . .	31

## Capítulo 3. Ambiente do Lotus Domino 33

Configurar o Lotus Domino para utilizar o LDAP do OS/400. . . . .	33
Detalhes da aplicação . . . . .	34
Pontos de concepção de aplicações . . . . .	34
Instalação da aplicação . . . . .	35
Formulários e vistas do Lotus Domino . . . . .	35
Agentes do Lotus Domino . . . . .	36
Observações do Lotus Domino . . . . .	37



---

# Capítulo 1. Descrição geral do cenário de PME da iDrink e de Cola Connections

Este relatório documenta a experiência da equipa de teste de soluções para clientes do IBM<sup>(R)</sup> eServer<sup>(TM)</sup> i5 de implementação de um cenário de pequena e média empresa (PME) utilizando o WebSphere<sup>(R)</sup> Application Server - Express e o Lotus<sup>(R)</sup> Domino<sup>(R)</sup>. Neste cenário, é utilizado o WebSphere Application Server - Express for iSeries<sup>(TM)</sup> para estabelecer uma presença inicial na Web através de páginas da Web dinâmicas, acesso a base de dados e acesso ao servidor de directório. Uma aplicação do Lotus Domino entretanto fornece páginas da Web dinâmicas internas para empregados da iDrink.

O cenário de PME da iDrink simula uma empresa de distribuição de bebidas. Esta empresa fictícia tem menos de duzentos empregados e é considerada uma pequena e média empresa. O distribuidor de bebidas contacta com os fornecedores de bebidas para obter o inventário dos produtos que vendem. A iDrink contacta com os clientes que compram grandes quantidades de bebidas para vender numa variedade de estabelecimentos. Os empregados contactam com os fornecedores e os clientes de modo a garantir o sucesso do negócio. A empresa iDrink processa as informações dos clientes, fornecedores e produtos através de servlets Java<sup>(TM)</sup> e JSPs (JavaServer Pages). A maioria das informações sobre clientes, fornecedores e produtos são armazenadas numa DB2 Universal Database<sup>(TM)</sup> com informações e palavras-passe dos empregados, palavras-passe dos clientes e palavras-passe dos fornecedores mantidas num directório LDAP (Lightweight Directory Access Protocol).

Além disso, os empregados da iDrink têm acesso a um sítio da Web de anúncios classificados com o nome Cola Connections que podem utilizar para anunciar itens para venda. Esta aplicação é implementada no Lotus Domino.

---

## Descrição geral da iDrink

A empresa iDrink permite que os empregados, clientes e fornecedores acedam a informações adequadas às funções.

### Decisões de concepção do cenário

Segue-se uma descrição das decisões de concepção do cenário tomadas inicialmente:

- Utilizar o WebSphere<sup>(R)</sup> Application Server - Express para o desenvolvimento de aplicações uma vez que este fornece um meio para estabelecer rapidamente uma presença na Web para os clientes, empregados e fornecedores à interface.
- Utilizar o directório eServer<sup>(TM)</sup> i5 LDAP para autenticação da aplicação WebSphere como segurança.

### Aplicação iDrink

A aplicação iDrink consiste num ambiente do WebSphere Application Server - Express utilizando JavaServer Pages, JavaBeans<sup>(TM)</sup> e servlets Java<sup>(TM)</sup> para permitir que os clientes, empregados e fornecedores executem as seguintes funções:

- Clientes
  - Registar ou iniciar sessão
  - Visualizar informações dos produtos
  - Encomendar produtos
  - Actualizar informações dos clientes
  - Visualizar histórico das encomendas
- Empregados

- Actualizar informações dos fornecedores
- Encomendar produtos aos fornecedores
- Modificar a fixação de preços dos produtos
- Registrar encomendas do cliente
- Registrar o inventário de produtos
- Criar, actualizar e visualizar anúncios classificados
- Fornecedores
  - Actualizar informações dos produtos

A figura 1 mostra o fluxo da aplicação iDrink.

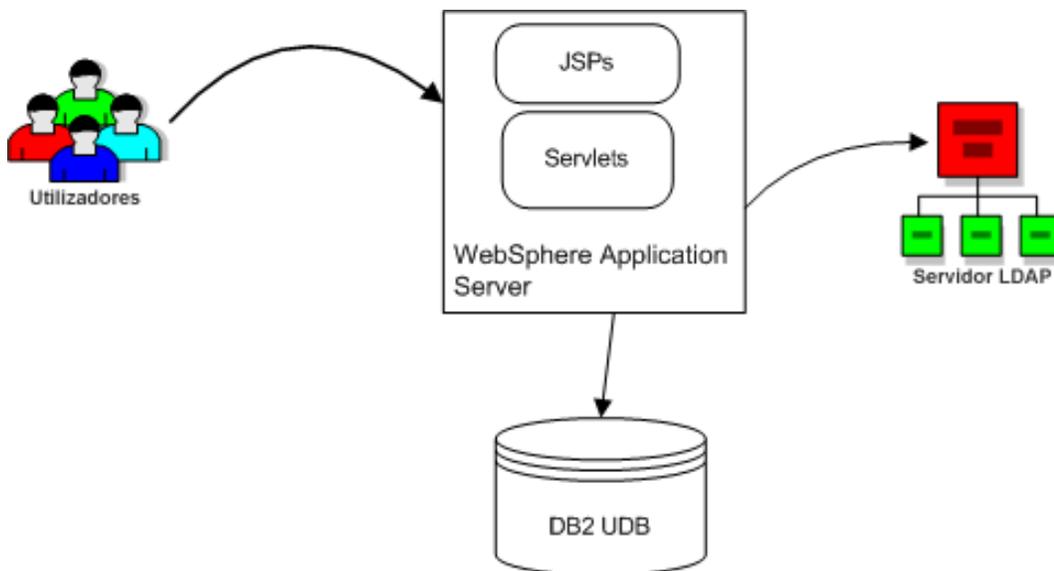


Figura 1: fluxo da aplicação iDrink

---

## Descrição geral de Cola Connections

A aplicação Cola Connections fornece aos empregados uma localização para divulgação de anúncios classificados a outros empregados da Drink.

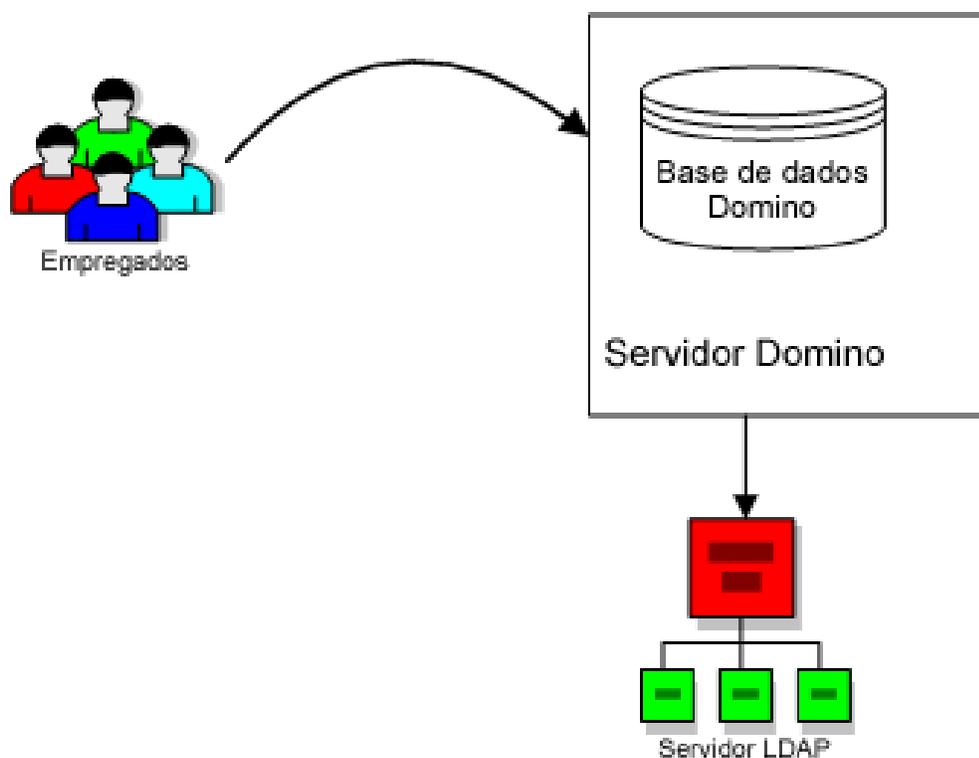
### Decisões de concepção de aplicações

Segue-se uma descrição das decisões de concepção tomadas inicialmente:

- Utilizar o Lotus<sup>(R)</sup> Domino<sup>(R)</sup> para o desenvolvimento de aplicações uma vez que fornece um meio de estabelecer uma rápida presença na Web.
- Utilizar o directório eServer<sup>(TM)</sup> i5 LDAP para autenticação de aplicações do Lotus Domino, como segurança.

### Aplicação Cola Connections

A aplicação Cola Connections é constituída por uma base de dados, agentes, vistas e formatos do Lotus Domino que permitem aos empregados da iDrink criar, actualizar ou visualizar anúncios classificados. A figura 2 apresenta o fluxo da aplicação Cola Connections.



**Figura 2: fluxo da aplicação Cola Connections**

### **Segurança utilizando a assistência de directório do Domino**

O directório do Domino fornece assistência para proteger o acesso a aplicações e dados. O mecanismo de segurança determina e verifica a identidade do utilizador e permite o acesso a recursos protegidos apenas a utilizadores designados. Na aplicação Cola Connections, só os empregados estão autorizados a aceder aos anúncios classificados. O Domino utiliza um modelo de controlo de acesso que armazena entradas na lista de controlo de acesso (ACL - Access Control List) para cada base de dados. A segurança é implementada através da configuração da assistência de directório do Domino para utilizar o directório i5/OS<sup>(TM)</sup> LDAP existente. Poderá obter detalhes sobre a configuração de segurança na secção Capítulo 3, “Ambiente do Lotus Domino”, na página 33.

---

## **Descrição geral do ambiente**

A configuração do ambiente de PME do cenário da iDrink é descrita nos seguintes tópicos: ambiente de tempo de execução, hardware do sistema e os principais produtos utilizados.

### **Ambiente de tempo de execução**

A empresa iDrink é considerada uma pequena e média empresa, por conseguinte adquiriu um eServer<sup>(TM)</sup> i5 acessível, de nível base, concebido para responder aos desafios internos. Este pequeno servidor permite a execução de uma variedade de aplicações da iDrink ao mesmo tempo, proporcionando segurança e disponibilidade, juntamente com a capacidade de execução das respectivas aplicações empresariais e soluções e-business.

### **Hardware do sistema**

É utilizado um sistema eServer i5 nesta implementação de cenário. Este sistema eServer i5 contém 1 processador unidireccional, 1 GB de memória e 175 GB de memória principal com o i5/OS<sup>(TM)</sup> V5R3.

## Produtos principais

Os principais produtos de software utilizados neste cenário são:

- **WebSphere<sup>(R)</sup> Application Server - Express for iSeries<sup>(TM)</sup>**: um servidor de aplicações e uma ferramenta de implementação solidamente integrada que fornecem um ponto de entrada extremamente acessível para e-business destinado a empresas que criam sítios da Web dinâmicos. O WebSphere Application Server - Express suporta as especificações de JSPs (JavaServer Pages), servlets Java<sup>(TM)</sup> e serviços da Web. O WebSphere Application Server - Express permite a criação de sítios da Web estáticos e dinâmicos, acedendo a informações existentes em bases de dados e executando actualizações simples, ao mesmo tempo que permite também a criação e utilização dos serviços da Web. Responde às necessidades das pequenas e médias empresas por ter um custo reduzido, ser fácil de utilizar, representar uma solução imediata.

Poderá obter informações detalhadas sobre o IBM<sup>(R)</sup> WebSphere Application Server - Express for iSeries na Web em: <http://www.ibm.com/servers/eserver/series/software/websphere/wsappserver/express/indexexp51.html>.

- **Lotus<sup>(R)</sup> Domino<sup>(R)</sup>**: fornece os princípios para colaboração e e-business, soluções base desde o sistema de mensagens empresariais a transacções baseadas na Web e tudo o que medeia. Este sistema de mensagens e colaboração de classe empresarial é criado para maximizar a produtividade destacando a experiência e conhecimentos de indivíduos, equipas e comunidades alargadas.

Poderá obter informações detalhadas sobre o Lotus Domino for iSeries na Web em: <http://www.ibm.com/servers/eserver/series/domino/>

- **LDAP (Lightweight Directory Access Protocol)**: um protocolo de serviço de directório que utiliza TCP/IP. O serviço de directório LDAP segue um modelo de cliente/servidor. Um ou mais servidores LDAP contêm os dados de directório. Um cliente LDAP liga a um servidor LDAP e efectua um pedido. O servidor apresenta uma resposta ou um apontador (uma referência) para outro servidor LDAP. Uma vez que o LDAP é considerado mais um serviço de directório do que uma base de dados, as informações num directório LDAP são normalmente descritíveis, baseadas em atributos. É muito mais frequente O LDAP ler as informações existentes no directório do que alterá-las. As actualizações são normalmente simples, muda tudo ou nada. As utilizações mais comuns dos directórios LDAP incluem:
  - listas telefónicas online
  - directórios de correio electrónico
  - autorização
  - autenticação

Poderá obter informações detalhadas sobre o LDAP na Web em: <http://publib.boulder.ibm.com/pubs/html/as400/infocenter.htm>

- **IBM HTTP Server suportado por Apache for iSeries**: um servidor da Web que fornece ferramentas para estabelecer rápida e facilmente uma presença na Web permitindo explorar a Web com objectivos comerciais.

Poderá obter informações detalhadas sobre o HTTP Server suportado por Apache for iSeries na Web em: <http://www.ibm.com/servers/eserver/series/software/http/>

- **DB2 Universal Database<sup>(TM)</sup> (UDB) for iSeries**: um sistema de gestão de bases de dados relacionais avançado, de 64 bits, que optimiza as funcionalidades On Demand do eServer i5 da IBM. Um membro da família de produtos líderes de mercado DB2<sup>(R)</sup> da IBM, o DB2 UDB for iSeries suporta uma vasta gama de aplicações e ambientes de desenvolvimento de baixo custo de propriedade, devido às funcionalidades únicas de computação autónoma (gestão automática).

Poderá obter informações detalhadas sobre o DB2 UDB na Web em: <http://www.ibm.com/servers/eserver/series/db2>

---

## Descobertas de PME

Segue-se uma lista das descobertas chave que foram reveladas durante a criação da solução de PME da iDrink:

- Se decidir armazenar informações nos dados de sessão para comunicação de dados entre servlets e JSPs, certifique-se de que os objectos armazenados têm um tamanho reduzido. Quaisquer objectos armazenados na sessão passam a fazer parte da pilha Java<sup>(TM)</sup> e requerem utilização de memória. Numa pequena caixa com pouca memória, não existe muito espaço para armazenar objectos. Se ultrapassar a atribuição de memória do sistema, tudo o que estiver na memória será paginado para o disco, diminuindo o desempenho global do sistema.
- Aumente o tamanho do conjunto de ligações de origem dos dados do WebSphere<sup>(R)</sup> para processar o número de pedidos previstos no sítio da Web. Defina também o valor Aged Timeout de modo a que seja possível eliminar as ligações que não estão a ser utilizadas, permitindo uma utilização reduzida da memória. Para definir estes valores na WebSphere Administration Console (Consola de administração do WebSphere):
  - Expandir o sinal de adição junto a **Resources** (Recursos) no painel da esquerda e, em seguida, clique na ligação **JDBC Providers** (Fornecedores de JDBC). No painel principal, escolha o selector de opção **Server Scope** (Âmbito do servidor) e prima o botão **Apply** (Aplicar)
  - No painel principal, clique na ligação do fornecedor JDBC utilizada pela aplicação. Na parte inferior da página dos fornecedores JDBC nas propriedades adicionais prima a ligação **Data Sources** (Origens de dados)
  - No painel principal, clique na ligação da origem de dados utilizada pela aplicação. Na parte inferior da página da origem dos dados clique na ligação **Connection Pool** (Conjunto de ligações)
  - Na página do conjunto de ligações defina as ligações mínimas e máximas, bem como o valor Aged Timeout
  - Clique no botão **Apply** (Aplicar) e, em seguida, na ligação **Save** (Guardar) no canto superior esquerdo do ecrã. Na página seguinte, clique no botão **Save** (Guardar)
  - Reinicie o WebSphere Application Server
- Aumente o número de módulos no contentor da Web dentro da instância-objecto do WebSphere Application Server de modo a corresponder ao número máximo de utilizadores. Do mesmo modo, clique para permitir a atribuição de módulos além do tamanho máximo estabelecido. Para definir estes valores na WebSphere Administration Console (Consola de administração do WebSphere):
  - Expandir o sinal de adição junto a **Servers** (Servidores) no painel da esquerda e, em seguida, clique na ligação **Application Servers** (Servidores da aplicação). No painel principal, clique na ligação do servidor de aplicação em que está a ser executada a aplicação
  - Na página dos servidores da aplicação, clique na ligação **Web Container** (Contentor da Web)
  - Na página do contentor da Web, clique na ligação **Thread Pool** (Conjunto de módulos)
  - Nesta página defina o tamanho mínimo e máximo ao mesmo tempo que faz clique na caixa de opção para **Is Growable** (Incrementável)
  - Clique no botão **Apply** (Aplicar) e, em seguida, na ligação **Save** (Guardar) no canto superior esquerdo do ecrã. Na página seguinte, clique no botão **Save** (Guardar)
  - Reinicie o WebSphere Application Server
- Aumente o número de módulos para processar pedidos no servidor de HTTP de modo a corresponder ao número máximo de utilizadores. Para definir estes valores na página de tarefas da IBM<sup>(R)</sup>:
  - Clique no separador **Manage** (Gerir) e, em seguida, clique no separador **HTTP Servers** (Servidores de HTTP)
  - Seleccione o servidor de HTTP associado ao WebSphere Application Server a partir da caixa pendente
  - No painel da esquerda seleccione a ligação **System Resources** (Recursos do sistema)
  - No painel principal, clique no separador **Advanced** (Avançadas)
  - Defina o número de módulos para processar pedidos
  - Prima o botão **Apply** (Aplicar) e, em seguida, prima o botão **OK**
  - Reinicie o servidor de HTTP

- Aumente os tamanhos das memórias tampão de envio e recepção de TCP do sistema consoante a natureza das aplicações em execução numa partição. Para alterar estas definições numa sessão 5250:
  - **CHGTCPA**
  - Altere os tamanhos das memórias tampão de recepção e envio de TCP
  - Prima **Enter**

---

## Capítulo 2. Ambiente do WebSphere Application Server - Express

O IBM<sup>(R)</sup> WebSphere Application Server - Express for iSeries<sup>(TM)</sup>, versão 5.1 é uma combinação das ferramentas de programação e de um servidor de aplicações que fornece um pacote integrado de aplicações baseadas na Web. O WebSphere Application Server - Express inclui o seguinte:

- Um servidor de aplicações com:
  - suporte Servlet 2.3 e JSP 1.2;
  - servidor da Web incorporado;
  - contentor da Web;
  - suporte de assistência da Web;
  - suporte XML e XSL;
  - suporte JDBC 2.0;
  - área de ligação;
  - Método de autenticação Websphere simples (SWAM, WebSphere Authentication Method);
  - suporte do Servlet 2.3 e JSP 1.2.
- O IBM WebSphere Development Studio Client for iSeries, versão 5.1, uma ferramenta de programação com:
  - especificação do Servlet 2.3;
  - especificação do JSP 1.2;
  - HTML;
  - JavaScript<sup>(TM)</sup> (do cliente e do servidor);
  - HTML Dinâmico (DHTML, Dynamic HTML);
  - XML e XHTML;
  - utilização e criação de serviços da Web;
  - programação em equipa com CVS;
  - JDBC 2.0;
  - suporte para a configuração e operação do servidor remoto;
  - bibliotecas de etiquetas de clientes;
  - Struts.

A aplicação iDrink está incluída num ficheiro Enterprise Archive (EAR). Um ficheiro EAR é um ficheiro JAR compactado que contém uma aplicação J2EE. As aplicações J2EE contêm elementos como servlets, JSPs, JavaBeans<sup>(TM)</sup> e ficheiros de configuração XML. O ficheiro EAR pode ser criado com o WebSphere Development Studio Client, o WebSphere Studio Application Developer ou o WebSphere Studio Application Site Developer.

---

### Descrição geral do WebSphere<sup>(R)</sup> Application Server

O ambiente WebSphere Application Server - Express foi configurado durante o trabalho de elaboração do cenário da iDrink Teste de soluções para clientes para o eServer<sup>(TM)</sup> i5. O objectivo era estabelecer rapidamente uma presença na Web para que a empresa iDrink pudesse fornecer uma interface aos seus clientes e fornecedores para executar as funções empresariais necessárias. A empresa utiliza também o ambiente do WebSphere Application Server - Express para fornecer uma interface para empregados executarem o seu trabalho. A interface da iDrink foi construída utilizando JavaServer Pages (JSPs), JavaBeans<sup>(TM)</sup> e servlets.

## Fluxo da aplicação

Existem vários JSPs utilizados na aplicação iDrink. Estes JSPs são apresentados e descritos nas secções Utilizador, Cliente, Empregado e Fornecedor.

O diagrama seguinte ilustra o grande fluxo da aplicação iDrink:

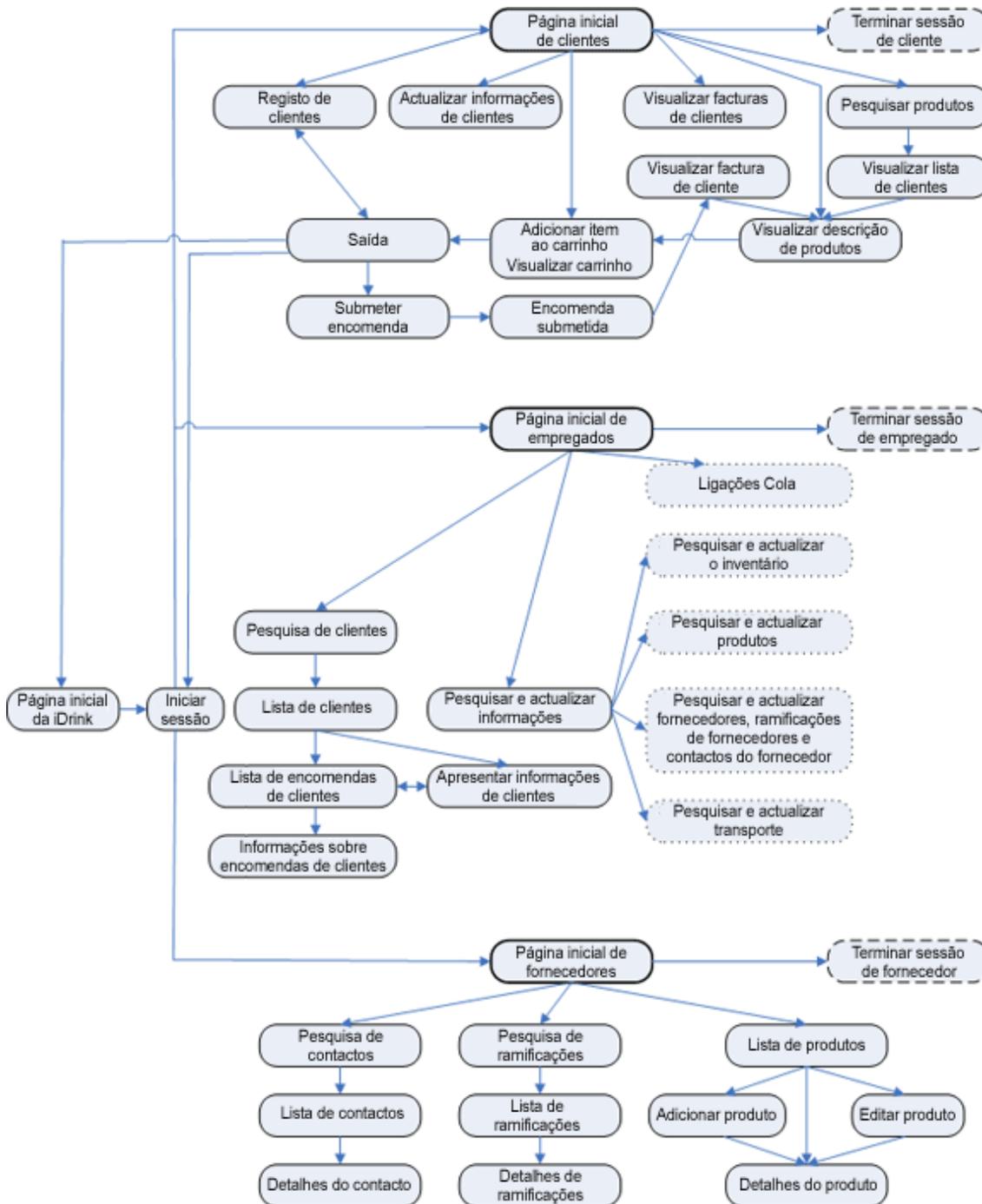


Figura 3: Fluxo da aplicação iDrink

---

## Ambiente de programação

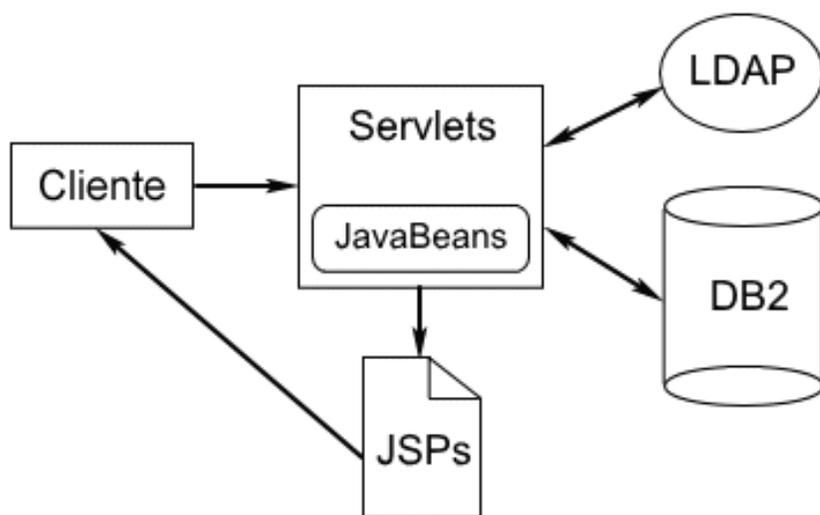
A equipa da iDrink escolheu o IBM<sup>(R)</sup> WebSphere<sup>(R)</sup> Studio Application Developer (WSAD) versão 5.0 para programar os servlets, JavaServer Pages (JSPs) e JavaBeans<sup>(TM)</sup> que em conjunto formam a aplicação iDrink. O WSAD oferece um ambiente de programação baseado em Eclipse que permite aos programadores criar, compilar e empacotar facilmente aplicações de Plataforma Java<sup>(TM)</sup> 2, Enterprise Edition (J2EE) para implementação num sistema de produção. Quando uma aplicação é codificada e compilada, pode ser executada num ambiente de teste no WSAD ou pode ser implementada numa instância do IBM WebSphere Application Server e configurada através da consola de administração do WebSphere. Outra ferramenta que pode ser utilizada para a implementação de aplicações da Web é o WebSphere Development Studio Client (WDSC).

---

## Detalhes da aplicação

A aplicação iDrink é uma aplicação para empresas composta por servlets Java<sup>(TM)</sup>, JavaBeans<sup>(TM)</sup> e JSPs (JavaServer Pages). Existem quatro servlets na aplicação iDrink. Estes incluem o UserServlet, o CustomerServlet, o EmployeeServlet e o SupplierServlet. O UserServlet é responsável pelo processamento de todos os pedidos de início de sessão, conclusão de sessão e palavra-passe de todos os utilizadores. O CustomerServlet é responsável pelo processamento de todos os pedidos dos clientes e de todos os utilizadores que não tenham sessão iniciada na aplicação iDrink. O EmployeeServlet processa todos os pedidos de empregados da iDrink que tenham iniciado sessão e o SupplierServlet processa todos os pedidos dos fornecedores da iDrink que tenham iniciado sessão.

Quando cada servlet da iDrink recebe um pedido de um utilizador, analisa o conteúdo do pedido para determinar a acção a efectuar. Por exemplo, o servlet pode ter de aceder ou criar uma entrada Lightweight Directory Access Protocol (LDAP) ou um registo num ficheiro de base de dados. Se os dados foram obtidos, é criado o JavaBean apropriado e os dados são armazenados no JavaBean. O JavaBean é então adicionado ao objecto do pedido ou ao objecto da sessão e o JSP apropriado é apresentado. A figura 4 ilustra a forma como estes componentes trabalham em conjunto.



**Figura 4: Componentes da aplicação iDrink**

Esta secção contém informações detalhadas sobre os métodos no UserServlet, no CustomerServlet, no EmployeeServlet e no SupplierServlet, para além dos JSPs chamados por estes servlets. Esta secção

contém também uma descrição dos pontos de concepção de aplicações que os servlets seguem, bem como os passos para instalar uma aplicação para empresas como a iDrink, num servidor IBM<sup>(R)</sup> WebSphere<sup>(R)</sup> Application Server - Express.

## Pontos de concepção de aplicações

Antes de conceber a aplicação, é tida em consideração a forma como o servidor LDAP e o esquema da base de dados estão concebidos. Esta secção contém as informações sobre estas considerações, bem como as concepções dos servlets JavaBeans<sup>(TM)</sup> e Java<sup>(TM)</sup> para o cenário iDrink.

### Concepção Lightweight Directory Access Protocol (LDAP)

Um directório LDAP é utilizado nas aplicações iDrink e Cola Connections como um mecanismo de autenticação. A figura 5 mostra como o directório LDAP é configurado.

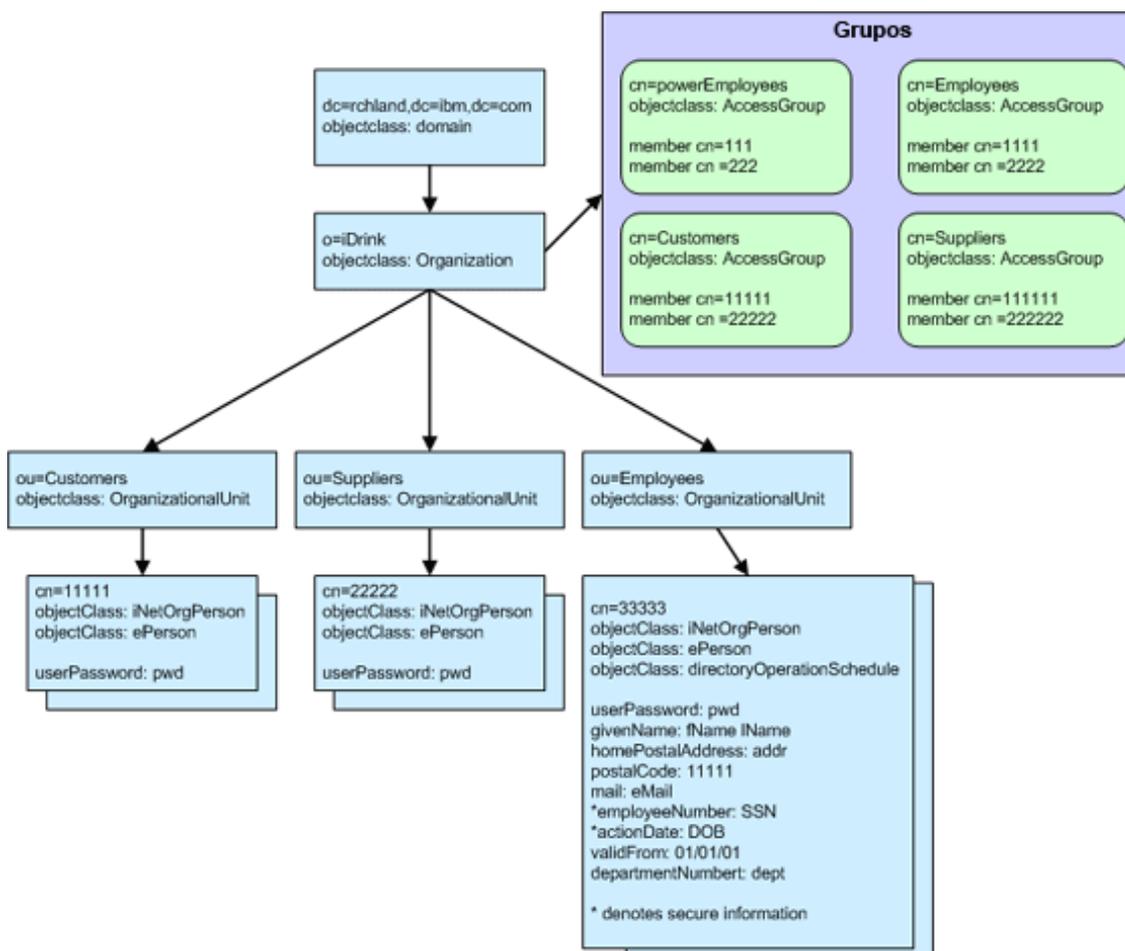


Figura 5: Directório LDAP

Quando configurar o directório LDAP, são utilizadas as seguintes opções de classes de objectos e ALCs:

- Os Clientes e os Fornecedores utilizam as classes de objectos inetOrgPerson e ePerson. Todos os clientes são membros de um grupo chamado cn=customers e todos os fornecedores são membros de um grupo chamado cn=suppliers.
- Os empregados utilizam as classes de objectos inetOrgPerson e ePerson. Todos os empregados são membros de um grupo chamado cn=employees. Alguns empregados são membros de um grupo

chamado `cn=powerEmployees`. O grupo `powerEmployees` tem permissão para efectuar determinadas acções nas entradas LDAP que os empregados comuns não têm permissão para efectuar.

- São adicionados filtros da Lista de controlo de acesso (ACL, Access Control List) aos registos de empregados para proteger os dados contidos em `actionDate` (Data de nascimento) e `employeeNumber` (Número da Segurança Social). Os filtros da ACL são utilizados para permitir aos empregados ler o registo, no entanto, apenas `powerEmployees` podem visualizar os valores de `employeeNumber` e `actionDate`.
  - `acentry: group:cn=employees,o=iDrink,dc=Nome domínio,dc=Sufixodomínio:normal:grant:rsc`
  - `acentry: group:cn=employees,o=iDrink,dc=Nome domínio,dc=Sufixodomínio:at.employeeNumber:deny:rsc:at.actionDate:deny:rsc`
  - `acentry: group:cn=powerEmployees,o=iDrink,dc=Nome domínio,dc=Sufixodomínio.:at.employeeNumber:grant:rsc:at.actionDate:grant:rsc`
- Quando preencher o directório LDAP, todos os campos de data têm um formato `AAAAMMDDHHMMSS[. |,fraction][(+ | -HHMM) | Z]`

### Concepção da base de dados

O seguinte diagrama ilustra a concepção da base de dados da `iDrink`:

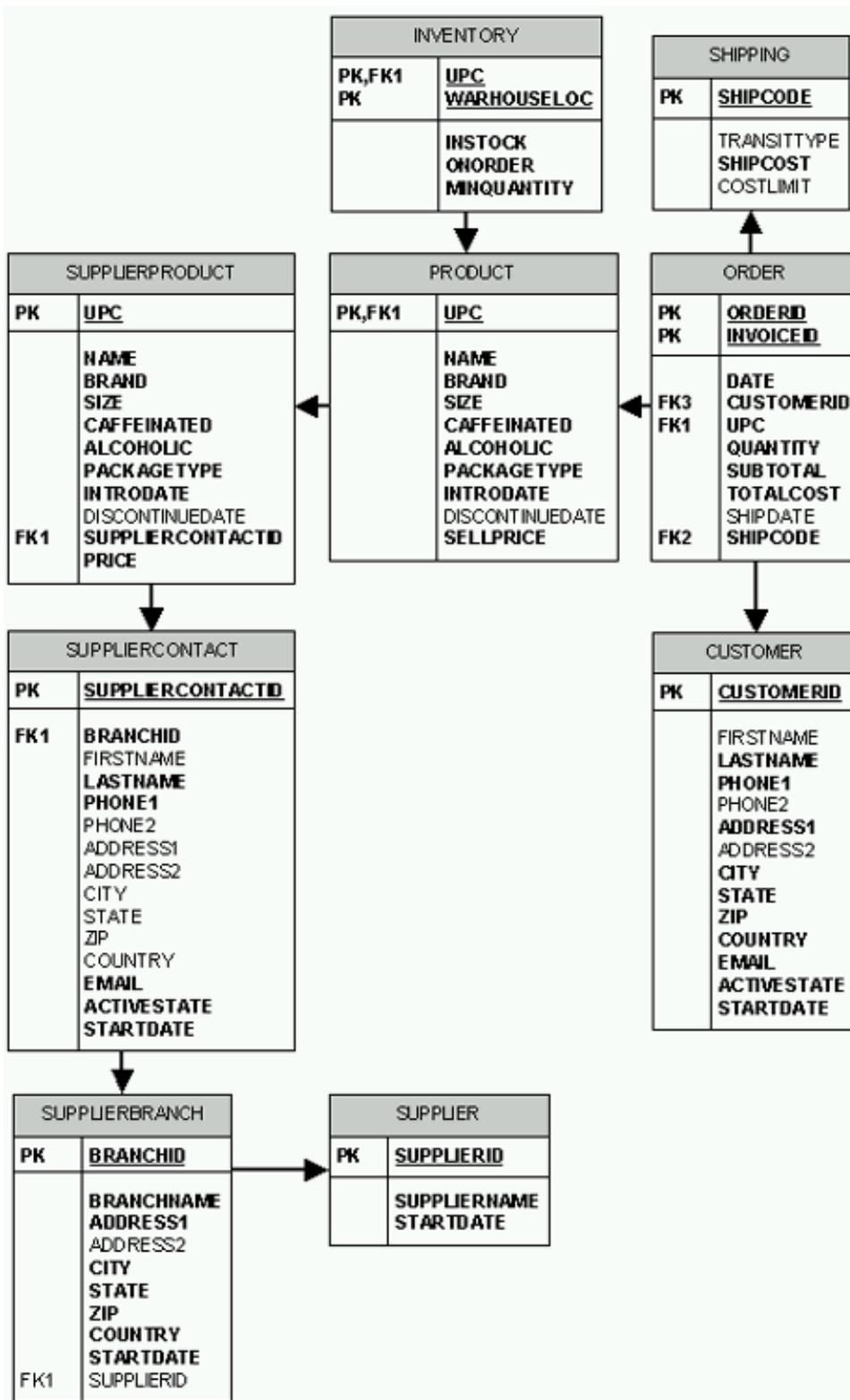


Figura 6: Concepção da base de dados da iDrink

A seguir é apresentado um breve resumo de cada tabela:

- SUPPLIER: Contém informações sobre os fornecedores da iDrink.

- **SUPPLIERBRANCH:** Contém informações sobre ramificações de fornecedores da iDrink.
- **SUPPLIERCONTACT:** Contém informações sobre contactos do fornecedor da iDrink.
- **SUPPLIERPRODUCT:** Contém informações sobre produtos fornecidos por fornecedores da iDrink's. Alguns dos produtos nesta base de dados não são vendidos pela iDrink.
- **PRODUCT:** Contém informações sobre produtos fornecidos pela iDrink. Os valores na tabela **PRODUCT** são um subconjunto dos valores na tabela **SUPPLIERPRODUCT**.
- **INVENTORY:** Contém informações sobre a quantidade de produtos que a iDrink tem actualmente e onde estes produtos estão localizados.
- **CUSTOMER:** Contém clientes que compram produtos da iDrink.
- **ORDER:** Contém informações de encomendas efectuadas por clientes da iDrink.
- **SHIPPING:** Contém os tipos de transporte utilizados pela iDrink para enviar encomendas.

### Concepção JavaBean

Os JavaBeans foram criados para armazenar dados obtidos pelos servlets. Na aplicação iDrink, foram criados os seguintes JavaBeans:

- **CustomerBean** - O CustomerBean armazena informações sobre a base de dados **CUSTOMER**.
- **CustomerInvoiceBean** - O CustomerInvoiceBean armazena informações da base de dados **ORDER**.
- **InventoryBean** - O InventoryBean armazena informações sobre a base de dados **INVENTORY**.
- **InvoiceItemBean** - O InvoiceItemBean armazena informações sobre um único artigo numa encomenda específica na base de dados **ORDER**.
- **ProductBean** - O ProductBean armazena informações sobre a base de dados **PRODUCT**.
- **ShippingBean** - O ShippingBean armazena informações sobre a base de dados **SHIPPING**.
- **SupplierBean** - O SupplierBean armazena informações sobre a base de dados **SUPPLIER**.
- **SupplierBranchBean** - O SupplierBranchBean armazena informações sobre a base de dados **SUPPLIERBRANCH**.
- **SupplierContactBean** - O SupplierContactBean armazena informações sobre a base de dados **SUPPLIERCONTACT**.
- **SupplierProductBean** - O SupplierProductBean armazena informações sobre a base de dados **SUPPLIERPRODUCT**.
- **UserBean** - O UserBean armazena informações do utilizador que se encontram em LDAP.

### Concepção do servlet

Foram utilizados quatro servlets na aplicação iDrink. A seguir é descrito brevemente cada um dos servlets:

- **CustomerServlet** - O CustomerServlet contém métodos utilizados no fluxo da aplicação de clientes.
- **EmployeeServlet** - O EmployeeServlet contém métodos utilizados no fluxo da aplicação de empregados. Alguns métodos são também utilizados no fluxo da aplicação de fornecedores.
- **SupplierServlet** - O SupplierServlet contém métodos utilizados no fluxo da aplicação de fornecedores.
- **UserServlet** - O UserServlet contém métodos utilizados em todo o fluxo da aplicação para assegurar que apenas os utilizadores com a autoridade correcta podem iniciar sessão e aceder às páginas apropriadas (por exemplo, impede que os clientes visualizem as páginas dos empregados).

---

## UserServlet

O UserServlet da iDrink fornece a funcionalidade de que qualquer utilizador da iDrink necessita para iniciar sessão na iDrink, para alterar a palavra-passe e para concluir sessão da iDrink. O servlet utiliza um directório LDAP para executar estas tarefas.

### Pontos de concepção

A aplicação iDrink necessitava de um mecanismo para determinar se um utilizador tinha sessão iniciada. Se um utilizador tinha sessão iniciada, a aplicação também necessitava de saber o tipo de utilizador. Para executar estas tarefas, a aplicação conta com a classe UserBean. Um objecto UserBean contém o ID do utilizador e um valor inteiro ao qual é atribuído um de três valores constantes que estão associados a cada um dos tipos de utilizador da iDrink. Quando um utilizador inicia sessão na iDrink, o UserServlet cria um novo objecto UserBean e define o ID do utilizador e os valores do tipo de utilizador e, em seguida, adiciona o objecto à sessão. O objecto permanece na sessão até que o utilizador conclui a sessão. Desta forma, quando um utilizador inicia sessão, qualquer servlet ou JSP acedido pelo mesmo pode facilmente determinar o ID do utilizador e o tipo de utilizador.

### Métodos

A seguir é apresentada uma lista de métodos que se encontram no UserServlet. O método doPost recebe todos os pedidos chegados do cliente e chama o método apropriado a partir desta lista de métodos.

- **assertUserType** - O método assertUserType obtém o UserBean do pedido. Se o UserBean for nulo ou não for de um tipo de utilizador válido, o método apresenta o JSP UserLogin.
- **changePassword** - O método changePassword aceita um ID do utilizador, uma palavra-passe antiga e uma nova palavra-passe. É localizada a entrada LDAP para o ID do utilizador especificado e o atributo userPassword nessa entrada é actualizado com a nova palavra-passe. Se esta operação for bem sucedida, o método devolve um valor booleano verdadeiro. Se ocorrerem quaisquer excepções durante este processo, o método devolve um valor booleano falso.
- **handlePasswdRequest** - O método handlePasswdRequest verifica se a nova palavra-passe corresponde à palavra-passe de confirmação e, em seguida, chama o método changePassword para alterar a palavra-passe existente. O método apresenta então o JSP UserPasswdResults.
- **loginUser** - O método loginUser obtém o ID do utilizador e a palavra-passe do pedido e chama o método verifyUser para verificar se a palavra-passe especificada está correcta para aquele ID do utilizador. Se a palavra-passe estiver correcta, é criado um UserBean e é adicionado ao objecto da sessão. O JSP apropriado é então apresentado, com base no tipo de utilizador que está a iniciar sessão. Um cliente verá o JSP iDrinkHome, um empregado verá o JSP EmployeeHome e um fornecedor verá o JSP SupplierHome.
- **logoutUser** - O método logoutUser remove o objecto UserBean do objecto da sessão e, em seguida, apresenta o JSP iDrinkHome.
- **verifyUser** - O método verifyUser aceita o ID do utilizador e a palavra-passe e verifica se a palavra-passe está correcta para o ID do utilizador especificado. Se a palavra-passe estiver correcta, o método devolve um número inteiro que representa o tipo de utilizador da iDrink. Se a palavra-passe estiver incorrecta, o método devolve zero.

### JavaServer Pages

A seguir é apresentada uma lista de JSPs que são chamados pelo UserServlet:

- **CustomerProceedToCheckout** - O JSP CustomerProceedToCheckout apresenta uma ligação que encaminhará um cliente ao JSP CustomerCheckout. Este JSP é utilizado como uma fase intermediária quando um cliente inicia sessão durante o processo de saída de sessão.
- **EmployeeHome** - O JSP EmployeeHome é apresentado depois de um empregado iniciar sessão.
- **iDrinkHome** - O JSP iDrinkHome é apresentado depois de um cliente iniciar sessão. É também apresentado depois de um utilizador concluir sessão.

- **SupplierHome** - O JSP SupplierHome é apresentado depois de um fornecedor iniciar sessão.
- **UserLogin** - O JSP UserLogin é apresentado no início do processo de início de sessão. Apresenta campos de texto para um ID do utilizador e uma palavra-passe.
- **UserPasswdResults** - A página UserPasswdResults é apresentada depois de um utilizador alterar a palavra-passe. Informa o utilizador se o pedido de alteração da palavra-passe foi ou não bem sucedido.

---

## CustomerServlet

O CustomerServlet da iDrink fornece a funcionalidade que um cliente da iDrink necessita para desenvolver relações comerciais com a iDrink. Dá a um cliente a capacidade de se registar na iDrink, actualizar as informações de cliente, pesquisar produtos, visualizar descrições de produtos, adicionar artigos ao carrinho de compras, efectuar encomendas e visualizar o histórico de encomendas. O CustomerServlet e os JSPs (JavaServer Pages) que chama trabalham com o UserServlet para fornecer a funcionalidade que permite a um utilizador iniciar e concluir sessão na iDrink e alterar a palavra-passe. Como valor assumido, o CustomerServlet trata todos os pedidos de um utilizador da iDrink, até ao momento em que inicia sessão como empregado ou fornecedor da iDrink. Se um utilizador da iDrink iniciar sessão como cliente da iDrink, o CustomerServlet continua a tratar os pedidos desse utilizador.

### Pontos de concepção

Quando um cliente efectua uma compra na iDrink, o CustomerServlet utiliza um objecto iDrinkCart para armazenar os artigos que um utilizador adiciona ao carrinho de compras. O objecto iDrinkCart contém um `java.util.Hashtable`, que armazena iDrinkCartItem adicionados ao iDrinkCart. Cada iDrinkCartItem contém informações sobre o artigo no carrinho, incluindo o número UPC, a quantidade e o preço. O valor chave de cada iDrinkCartItem na tabela de indexações é o número UPC desse artigo, o que permite uma obtenção mais fácil dos artigos na tabela de indexações.

Quando um novo cliente se regista na iDrink, deve ser gerado um novo ID de cliente único e atribuído à conta do novo utilizador. Só existe um registo na tabela de clientes para cada cliente, pelo que é utilizada uma coluna de identidade para gerar este valor do ID de cliente. Uma coluna de identidade fornece uma forma fácil de gerar automaticamente um valor chave primário único para cada linha numa tabela. Isto elimina a os problemas de concorrência e de desempenho que podem ocorrer quando uma aplicação tem de gerar os seus próprios valores únicos, normalmente com base nos valores já existentes na tabela. Uma vez que o ID de cliente é uma coluna de identidade na tabela de clientes, o CustomerServlet não tem de tentar determinar qual deve ser o ID de cliente seguinte. Deixa simplesmente a funcionalidade da coluna de identidade tratar disso automaticamente.

Quando um cliente efectua uma encomenda na iDrink, deve ser gerada um novo ID de factura único e atribuído a essa encomenda. Quando o CustomerServlet recebe as informações da encomenda do cliente, insere um registo na tabela de encomendas para cada artigo único na encomenda. Cada um destes registos deve conter o mesmo ID de factura. Desta forma, pode existir um número qualquer de registos na tabela de encomendas com o mesmo ID de factura. Por este motivo, não foi possível utilizar uma coluna de identidade para gerar um ID de factura único, uma vez que uma coluna de identidade geraria um valor único para cada artigo na encomenda, em vez de um valor único para toda a encomenda. Para gerar um ID de factura único, o CustomerServlet conta com um objecto de sequência DB2<sup>(R)</sup>. Um objecto de sequência DB2 é um objecto que gera valores sequenciais. Quando é pedido um valor a partir de um objecto de sequência, este devolve o valor seguinte numa sequência definida na altura em que o objecto de sequência foi criado. A sequência definida no objecto de sequência do ID de factura é apenas uma sequência crescente de números inteiros, com incremento de um. Quando o CustomerServlet efectua uma encomenda para um cliente, acede ao objecto de sequência para obter o ID de factura seguinte. Este valor é então utilizado em todas as instruções de inserção de SQL excluídas da encomenda que está a ser efectuada.

Os clientes não são obrigados a iniciar sessão antes de utilizar o sítio da Web da iDrink. Assim sendo, era necessário adicionar uma lógica ao CustomerServlet, ao UserServlet e aos vários JSPs a tratar no caso de

um cliente adicionar artigos ao carrinho e não iniciar sessão ou não se registar antes de iniciar o processo de saída. Se um cliente não tiver iniciado sessão antes de premir o botão de saída, é enviado um parâmetro adicional no pedido HTTP para o CustomerServlet (para registo do cliente) ou para o UserServlet (para início de sessão do utilizador). Se nenhum destes servlets encontrar este parâmetro num pedido de registo ou de início de sessão, o servlet processará o pedido e redireccionará o utilizador para uma página intermédia de saída que apresenta o ID de cliente e uma ligação para continuar o processo de saída. Se um cliente iniciar o processo de início de sessão ou de registo a partir de qualquer outro ponto do sítio da Web da iDrink, é redireccionado para um ecrã de boas-vindas, quando o pedido de início de sessão ou de registo estiver concluído com êxito. Neste caso, a página intermédia não é necessária e não é apresentada.

## Fluxo da aplicação

A interface do cliente da iDrink não impõe um caminho predeterminado a um cliente. Desta forma, um cliente pode facilmente passar de uma tarefa para outra em qualquer altura. Como consequência desta flexibilidade, é muito difícil ilustrar todos os caminhos possíveis que um cliente pode percorrer através da interface do cliente enquanto faz compras na iDrink. A figura 7 ilustra os caminhos utilizados com maior frequência pelos clientes da iDrink.

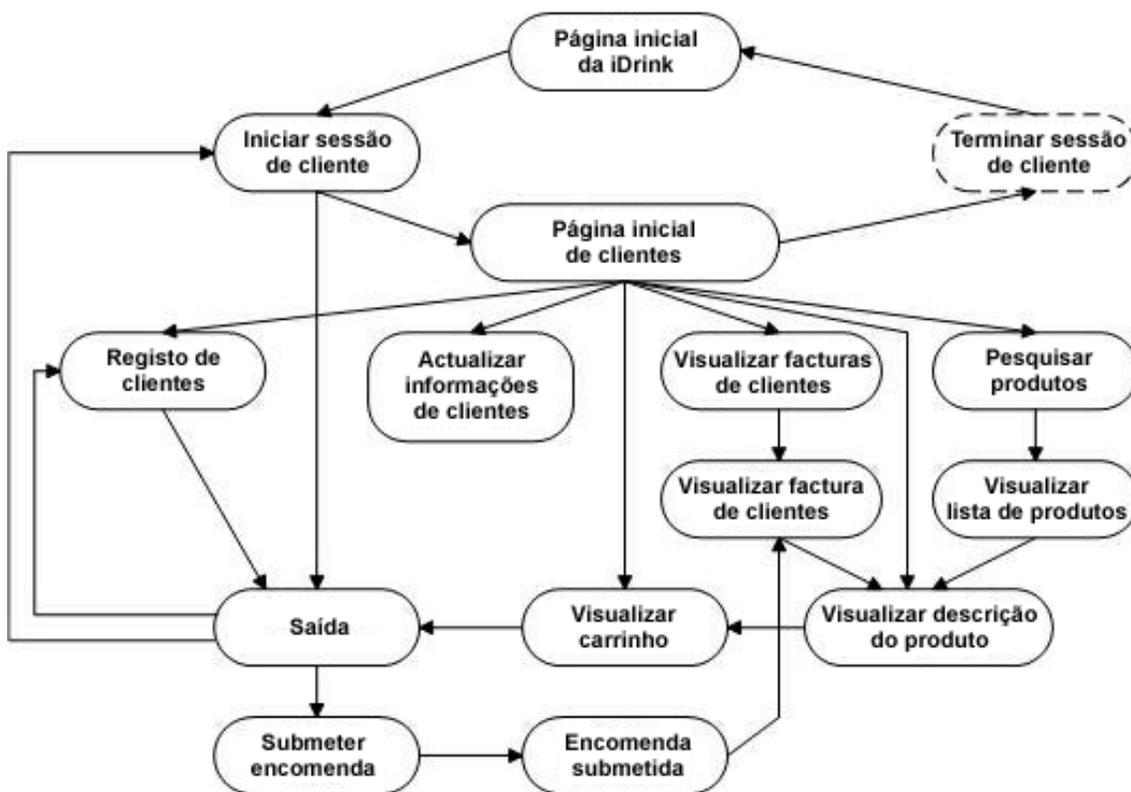


Figura 7: Fluxo da aplicação de clientes da iDrink

## Métodos

A seguir é apresentada uma lista de métodos que se encontram no CustomerServlet. O método doPost recebe todos os pedidos chegados do cliente e chama o método apropriado a partir desta lista de métodos.

- **addItemToCart** - O método addItemToCart adiciona uma quantidade especificada de um produto especificado ao iDrinkCart do cliente. O conteúdo actualizado do carrinho é apresentado no JSP do CustomerViewCart.

- **checkout** - O método checkout obtém os códigos de transporte válidos para uma encomenda do cliente, com base no valor dos artigos no respectivo iDrinkCart. Estes códigos de transporte são apresentados para serem seleccionados pelo cliente no JSP do CustomerCheckout.
- **displayMatchingProducts** - O método displayMatchingProducts obtém informações básicas sobre produtos que correspondem aos critérios de pesquisa especificados por um cliente. Estas informações são apresentadas no JSP CustomerMatchingProducts.
- **getNextInvoiceID** - O método getNextInvoiceID devolve o ID de factura seguinte, que é fornecido por um objecto de sequência DB2.
- **registerNewCustomer** - O método registerNewCustomer insere um novo registo na tabela de clientes da iDrink. Este registo contém as informações do cliente. Além disso, cria uma nova entrada LDAP para o cliente. Esta entrada contém a palavra-passe do cliente.
- **submitOrder** - O método submitOrder obtém o novo ID de factura a partir do método getNextInvoiceID. Em seguida, insere um registo na tabela de encomendas da iDrink para cada artigo no iDrinkCart do cliente.
- **updateCustomer** - O método updateCustomer actualiza as informações do cliente actual com valores fornecidos pelo cliente.
- **updateItemsInCart** - O método updateItemsInCart actualiza a quantidade de cada artigo no iDrinkCart do cliente, com base em valores fornecidos pelo cliente. O conteúdo actualizado é apresentado no JSP CustomerViewCart.
- **viewInvoices** - O método viewInvoices obtém informações básicas sobre cada encomenda que é efectuada pelo cliente actual. Estas informações são apresentadas no JSP CustomerInvoices.
- **viewInvoiceInformation** - O método viewInvoiceInformation obtém informações detalhadas sobre uma encomenda do cliente especificada. Estas informações são apresentadas no JSP CustomerInvoiceInfo.
- **viewProduct** - O método viewProduct obtém informações detalhadas sobre um produto através de um UPC especificado. Estas informações são apresentadas no CustomerProductInfo.
- **viewUpdateCustomerScreen** - O método viewUpdateCustomerScreen obtém informações do cliente actual. Estas informações são apresentadas no JSP CustomerUpdate.

## JavaServer Pages

A seguir é apresentada uma lista de JSPs que são chamados pelo CustomerServlet:

- **CustomerCheckout** - O JSP CustomerCheckout apresenta uma lista de opções de compra disponíveis que um cliente pode seleccionar para a sua encomenda.
- **CustomerDisplayCart** - O JSP CustomerDisplayCart apresenta o conteúdo do iDrinkCart do cliente.
- **CustomerInvoiceInfo** - O JSP CustomerInvoiceInfo apresenta informações detalhadas sobre uma encomenda do cliente específica.
- **CustomerInvoices** - O JSP CustomerInvoices apresenta uma lista de todas as encomendas que foram efectuadas por um cliente.
- **CustomerMatchingProducts** - O JSP CustomerMatchingProducts apresenta todos os produtos que correspondem aos critérios de pesquisa especificados por um cliente.
- **CustomerOrderSubmitted** - O JSP CustomerOrderSubmitted informa um utilizador que a sua encomenda foi efectuada e apresenta o número da factura.
- **CustomerProceedToCheckout** - O JSP CustomerProceedToCheckout apresenta uma ligação que encaminhará um cliente ao JSP CustomerCheckout. Este JSP é utilizado como uma fase intermediária quando um cliente se regista ou inicia sessão durante o processo de saída.
- **CustomerProductInfo** - O CustomerProductInfo apresenta informações detalhadas sobre um artigo da iDrink específico.
- **CustomerRegistration** - O JSP CustomerRegistration fornece campos de entrada de dados que permitem a um novo cliente introduzir e submeter as respectivas informações de cliente.
- **CustomerSearchProducts** - O JSP CustomerSearchProducts fornece campos de entrada de dados que permitem a um cliente introduzir e submeter um conjunto de critérios de pesquisa de produtos.

- **CustomerUpdate** - O JSP CustomerUpdate apresenta informações do cliente actuais nos campos de texto, o que permite a um utilizador modificar e submeter informações do cliente actualizadas.

---

## EmployeeServlet

O iDrink EmployeeServlet fornece a funcionalidade que um empregado da iDrink necessita para trabalhar com todas as entidades da empresa, incluindo clientes, fornecedores, encomendas, produtos e inventário. Oferece aos empregados a possibilidade de:

- visualizar as encomendas dos clientes;
- adicionar, visualizar, actualizar e eliminar produtos que a iDrink oferece;
- adicionar, visualizar, actualizar e eliminar produtos do inventário;
- criar, visualizar, actualizar e eliminar fornecedores, ramificações de clientes, e contactos do fornecedor;
- criar, visualizar, actualizar e eliminar tipos de transporte;
- aceder ao sítio da Cola Connections.

O EmployeeServlet e os JSPs que chama, trabalham com o UserServlet para fornecer a funcionalidade que garante que apenas os empregados podem iniciar sessão e aceder às páginas associadas ao EmployeeServlet.

### Pontos de concepção

- Além de utilizar colunas de identidade na concepção do fluxo da aplicação de clientes, as colunas de identidade são também utilizadas no fluxo da aplicação de empregados. As colunas de identidade são utilizadas nestas tabelas para permitir uma criação mais fácil de valores chave primários:
  - SHIPPING
  - SUPPLIER
  - SUPPLIERBRANCH
  - SUPPLIERCONTACT
- Comparando os fluxos das aplicações de empregados e de fornecedores, descobriu-se que estas acções eram semelhantes:
  - Pesquisar e visualizar informações do fornecedor.
  - Pesquisar e visualizar informações de ramificações de fornecedores.
  - Pesquisar e visualizar informações de contactos do fornecedor.

Uma vez que estas acções eram idênticas, foi decidido escrever apenas um conjunto de métodos para estas acções no EmployeeServlet em vez de escrever métodos no EmployeeServlet e no SupplierServlet.

Além disso, apenas foi criado um conjunto de páginas JSP para pesquisar e visualizar as informações. Uma vez que foi utilizado apenas um conjunto de JSPs, foi importante especificar que os fornecedores e os empregados tinham permissão para aceder a estas páginas. Isto foi feito através da adição da condição de verificação do fornecedor aos JSPs de pesquisa e visualização, como demonstra este código:

```
UserServlet.assertUserType
(request, response, UserBean.USER_TYPE_SUPPLIER | UserBean.USER_TYPE_EMPLOYEE, "UserLogin.jsp");
```

Uma vez que o fluxo da aplicação de empregados necessita que as ligações actualizem e eliminem o fornecedor, a ramificação de fornecedores e as informações de contactos do fornecedor a partir destas páginas, era importante verificar o tipo de utilizador antes de apresentar estas ligações. Se o utilizador não fosse do tipo fornecedor, as ligações para actualizar e eliminar não eram apresentadas. Este código JSP é um exemplo:

```
<%
  if(userbean.getUserType() == UserBean.USER_TYPE_EMPLOYEE)
  {
    %>

    <A HREF="EmployeeServlet?action=viewSupplierContactToUpdate&contactID=<%=
```

```

        supplierContactBean.getContactID()%>">Update Supplier Contact</A>
<br /><BR>
<A HREF="EmployeeServlet?action=viewSupplierContactToDelete&contactID=<%=
        supplierContactBean.getContactID()%>">Delete Supplier Contact</A>
<br /><BR>

<%
}

```

Quando um empregado elimina um fornecedor, o contacto do fornecedor ou o inventário, essa entidade é eliminada da base de dados. No entanto, a eliminação de produtos e de códigos de envio tem um significado diferente. Uma vez que é necessário manter as informações na base de dados para actualização do histórico, os artigos são deixados na base de dados e é definido um campo específico para representar que o artigo foi eliminado.

- Para eliminar um produto é definido o valor de DISCONTINUEDATE. (Um produto com um valor de DISCONTINUEDATE nulo significa que esse produto não foi eliminado).
- Para eliminar um código de envio, o valor de COSTLIMIT é definido como 0. (Um código de envio com um valor de COSTLIMIT maior do que 0 é um código de envio que pode ser utilizado para encomendas.)

Além disso, alguns artigos têm condições que devem ser cumpridas para a eliminar o artigo:

- Para eliminar um fornecedor, este não pode ter ramificações de fornecedores associadas a si.
- Para eliminar uma ramificação de fornecedores, o fornecedor não pode ter contactos de fornecedores associados à ramificação de fornecedores.
- Para eliminar um contacto do fornecedor, o contacto do fornecedor não pode ter produtos do fornecedor associados ao contacto do fornecedor e este não pode estar no estado activo.
- Para os empregados da iDrink adicionarem um novo produto às ofertas dos clientes, os empregados devem escolher um produto da tabela SUPPLIERPRODUCT que ainda não esteja na tabela PRODUCT. Lembre-se que a tabela SUPPLIERPRODUCT contém todos os produtos que são oferecidos pelos fornecedores da iDrink's. A tabela PRODUCT contém produtos que a iDrink oferece aos seus clientes. Os produtos na tabela PRODUCT são um subconjunto dos produtos na tabela SUPPLIERPRODUCT. Para apresentar os produtos na tabela SUPPLIERPROUDCT que não estão na tabela PRODUCT, foi executada uma selecção secundária na tabela SUPPLIERPRODUCT, como o demonstra o seguinte exemplo:

```

SELECT *
FROM IDRINK.SUPPLIERPRODUCT
WHERE UPC NOT IN (SELECT UPC FROM
                  IDRINK.PRODUCT)

```

- Durante a actualização do inventário existente, haviam várias opções que podiam ser implementadas:
  - Sobrepor o inventário existente por um novo montante.
  - Adicionar ou subtrair um montante do inventário existente.

Após muita consideração, foi decidido implementar a primeira opção: substituir o inventário existente por um novo montante. Esta era a opção mais simples para implementar e para utilizar pelos empregados.

- Durante o trabalho de transporte, é benéfico para os empregados visualizar os limites de custo mínimos e máximos de uma encomenda para cada código de envio. No entanto, na tabela SHIPPING, apenas o limite de custo máximo do código de envio é armazenado. O limite de custo mínimo está implícito pelo limite de custo máximo para o próximo limite de custo inferior para o tipo de transporte do código de envio (os tipos de transporte utilizados actualmente pela iDrink incluem TERRA, DIA SEGUINTE POR AR e SEGUNDO DIA POR AR). Para que os JSPs acedam facilmente à informação do limite de custo mínimo, foi criada uma variável chamada costLimitMin em ShippingBean para conter esta informação.

## Fluxo da aplicação

A interface do cliente da iDrink não impõe um caminho predeterminado aos empregados. Desta forma, um empregado pode facilmente passar de uma tarefa para outra em qualquer altura. Como consequência desta flexibilidade, é muito difícil ilustrar todos os caminhos possíveis que um empregado pode percorrer através da interface do cliente. A figura 8 ilustra o fluxo da aplicação básico disponível para os empregados da iDrink.

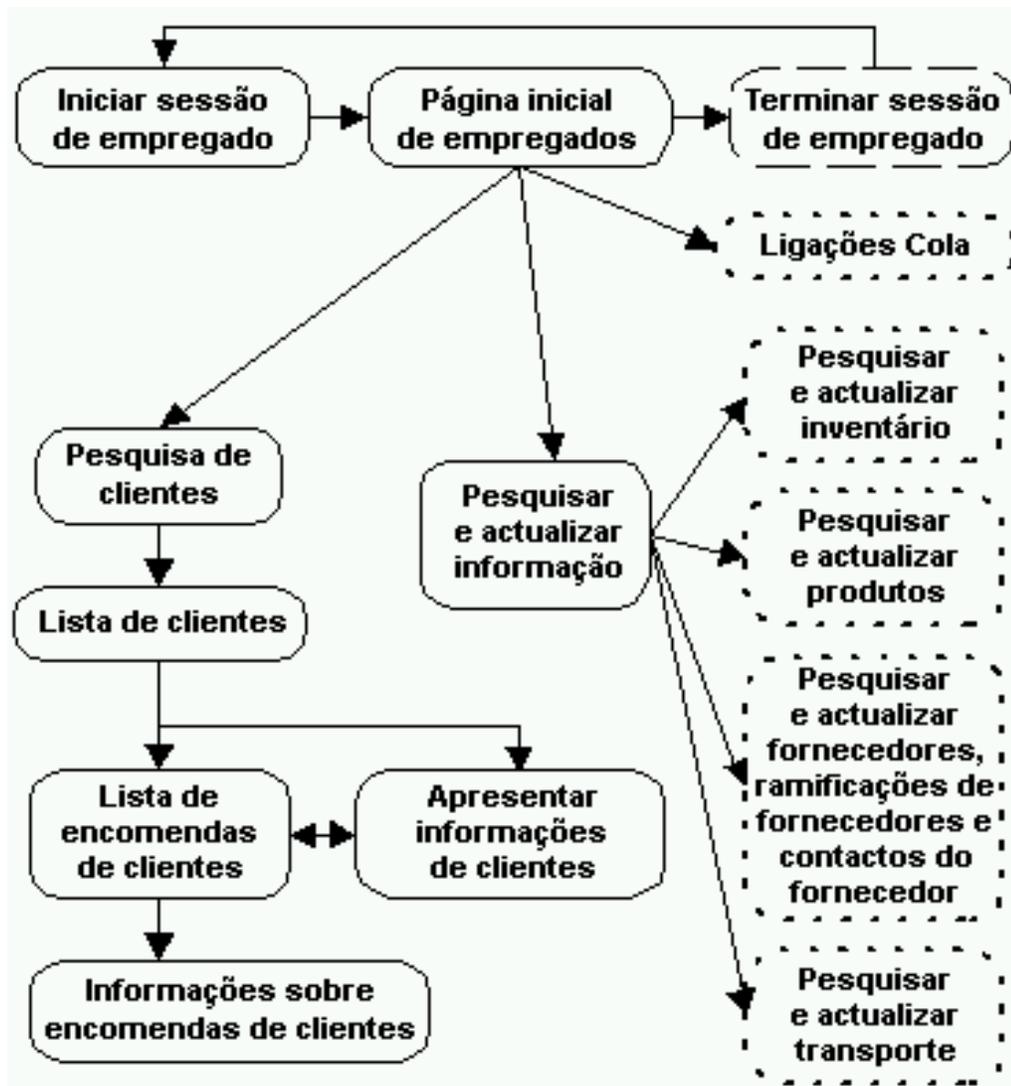


Figura 8: Fluxo da aplicação de empregados da iDrink

O fluxo da aplicação destes itens é semelhante:

- Pesquisar e actualizar o inventário.
- Pesquisar e actualizar produtos.
- Pesquisar e actualizar fornecedores, Ramificações de fornecedores e Contactos do fornecedor
- Pesquisar e actualizar transporte

Por exemplo, a figura 9 ilustra o fluxo da aplicação para Pesquisar e actualizar produtos:

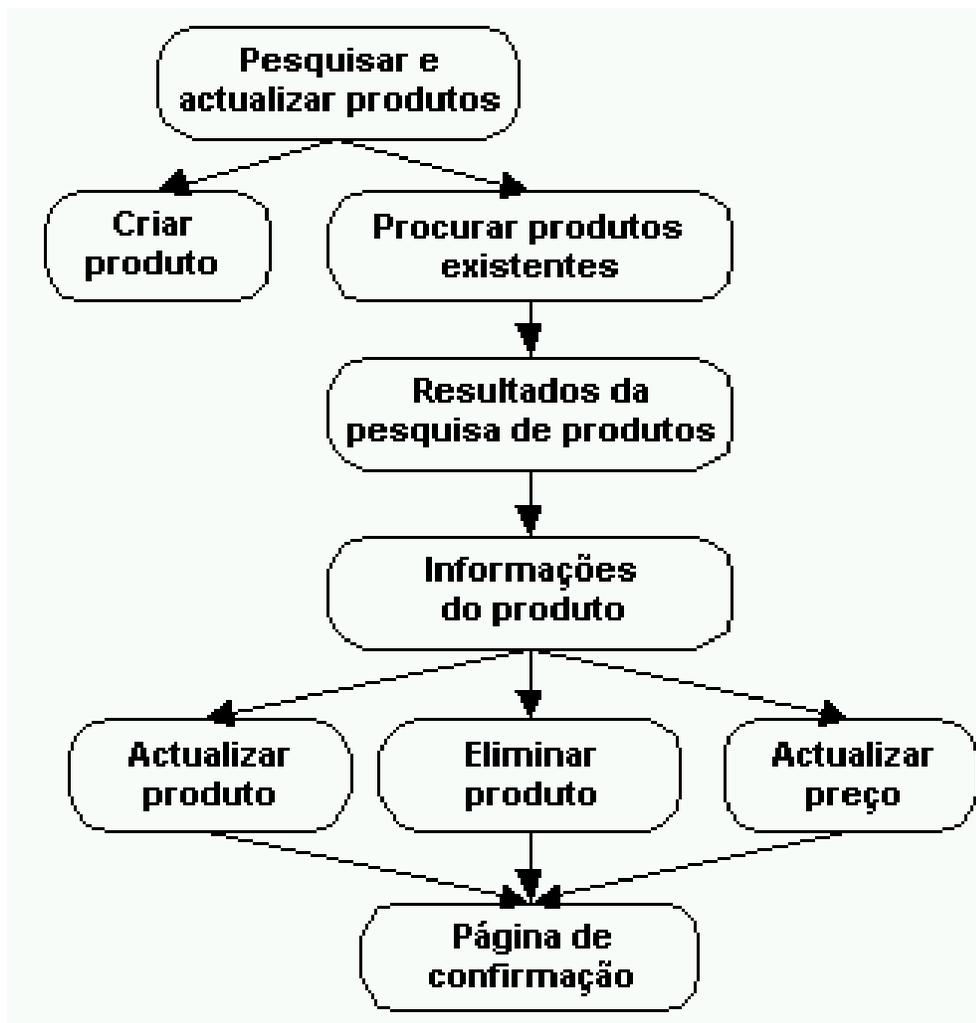


Figura 9: Fluxo da aplicação Pesquisar e actualizar produtos

Com algumas pequenas excepções, o mesmo fluxo da aplicação é utilizado para o Inventário, Transporte, Fornecedores, Ramificações de fornecedores e Contactos do fornecedor.

O fluxo da aplicação Cola Connections é descrito na secção Ambiente do Lotus<sup>(R)</sup> Domino<sup>(R)</sup>.

### Métodos

As seguintes tabelas contêm os métodos que se encontram em EmployeeServlet. Uma vez que a maior parte das entidades da iDrink têm métodos que executam acções semelhantes, os métodos são categorizados por acção.

O método doPost recebe todos os pedidos chegados do cliente e chama o método apropriado desta lista de métodos.

Entidade	Nome do método	Tabelas utilizadas	O que é chamado quando o método é executado
Inventário	createInventory	INVENTORY	Confirm.jsp
Produto	createProduct	PRODUCT, SUPPLIER PRODUCT	Confirm.jsp

Transporte	createShipping	SHIPPING	Confirm.jsp
Fornecedor	createSupplier	SUPPLIER	Confirm.jsp
Ramificação de fornecedores	createSupplierBranch	SUPPLIER, SUPPLIERBRANCH	Confirm.jsp
Contacto do fornecedor	createSupplierContact	SUPPLIER, SUPPLIERBRANCH, SUPPLIERCONTACT	Confirm.jsp

**Tabela 1: Métodos para criar um novo objecto e inserir o objecto na tabela apropriada.**

Entidade	Nome do método	Tabelas utilizadas	O que é chamado quando o método é executado
Inventário	deleteInventory	INVENTORY	Confirm.jsp
Produto	deleteProduct	PRODUCT	Confirm.jsp
Transporte	deleteShipping	SHIPPING	Confirm.jsp
Fornecedor	deleteSupplier	SUPPLIER	Confirm.jsp
Ramificação de fornecedores	deleteSupplierBranch	SUPPLIER, SUPPLIERBRANCH	Confirm.jsp
Contacto do fornecedor	deleteSupplierContact	SUPPLIER, SUPPLIERBRANCH, SUPPLIERCONTACT	Confirm.jsp

**Tabela 2: Métodos para permitir aos empregados remover objectos. Todos os métodos excepto deleteProduct e deleteShipping eliminarão o objecto da base de dados, desde que as condições apropriadas sejam cumpridas.**

Entidade	Nome do método	Tabelas utilizadas	O que é chamado quando o método é executado
Cliente	searchCustomerOrders	ORDER	EmployeeCustomerOrdersList.jsp
Cliente	searchCustomers	CUSTOMER	EmployeeCustomerList.jsp
Inventário	searchInventory	INVENTORY, PRODUCT	InventorySearchResults.jsp
Produto	searchProduct	PRODUCT	ProductSearchResults.jsp
Transporte	searchShipping	SHIPPING	ShippingSearchResults.jsp
Fornecedor	searchSupplier	SUPPLIER	SupplierSearchResults.jsp
Ramificação de fornecedores	searchSupplierBranch	SUPPLIER, SUPPLIERBRANCH	SupplierBranchSearchResults.jsp
Contacto do fornecedor	searchSupplierContact	SUPPLIER, SUPPLIERBRANCH, SUPPLIERCONTACT	SupplierContactSearchResults.jsp
Produto do fornecedor	searchSupplierProduct	PRODUCT, SUPPLIERCONTACT, SUPPLIERPRODUCT	ProductSearchResults.jsp

**Tabela 3: Métodos para permitir a um empregado pesquisar um objecto.**

Entidade	Nome do método	Tabelas utilizadas	O que é chamado quando o método é executado
Inventário	updateInventory	INVENTORY	Confirm.jsp

Produto	updatePrice	PRODUCT	Confirm.jsp
Produto	updateProduct	PRODUCT	Confirm.jsp
Transporte	updateShipping	SHIPPING	Confirm.jsp
Fornecedor	updateSupplier	SUPPLIER	Confirm.jsp
Ramificação de fornecedores	updateSupplierBranch	SUPPLIER, SUPPLIERBRANCH	Confirm.jsp
Contacto do fornecedor	updateSupplierContact	SUPPLIER, SUPPLIERBRANCH, SUPPLIERCONTACT	Confirm.jsp

**Tabela 4: Métodos para permitir a um empregado actualizar um objecto.**

Entidade	Nome do método	Tabelas utilizadas	O que é chamado quando o método é executado
Cliente	viewCustomer	CUSTOMER	EmployeeCustomerInfo.jsp
Cliente	viewCustomerOrder	ORDER, PRODUCT, SHIPPING	EmployeeOrderInfo.jsp
Inventário	viewInventoryItem	INVENTORY	InventoryInfo.jsp
Produto	viewPrice	PRODUCT	PriceUpdate.jsp
Produto	viewProductItem	PRODUCT	ProductInfo.jsp
Transporte	viewShippingItem	SHIPPING	ShippingInfo.jsp
Fornecedor	viewSupplierItem	SUPPLIER	SupplierInfo.jsp
Ramificação de fornecedores	viewSupplierBranchItem	SUPPLIER, SUPPLIERBRANCH	SupplierBranchInfo.jsp
Contacto do fornecedor	viewSupplierContactItem	SUPPLIER, SUPPLIERBRANCH, SUPPLIERCONTACT	SupplierContactInfo.jsp

**Tabela 5: Métodos para permitir a um empregado visualizar os detalhes de um objecto.**

Entidade	Nome do método	Tabelas utilizadas	O que é chamado quando o método é executado
Inventário	inventoryInfo	INVENTORY, PRODUCT	
Inventário	viewInventoryToDelete (chama inventoryInfo)		InventoryDelete.jsp
Inventário	viewInventoryToUpdate (chama inventoryInfo)		InventoryUpdate.jsp
Produto	productInfo	PRODUCT	
Produto	viewProductToDelete (chama productInfo)		ProductDelete.jsp
Produto	viewProductToUpdate (chama productInfo)		ProductUpdate.jsp
Transporte	fetchShippingInfo	SHIPPING	
Transporte	viewShippingToDelete (chama fetchShippingInfo)		ShippingDelete.jsp
Transporte	viewShippingToUpdate (chama fetchShippingInfo)		ShippingUpdate.jsp

Fornecedor	fetchSupplierInfo	SUPPLIER	
Fornecedor	viewSupplierToDelete (chama fetchSupplierInfo)		SupplierDelete.jsp
Fornecedor	viewSupplierToUpdate (chama fetchSupplierInfo)		SupplierUpdate.jsp
Ramificação de fornecedores	fetchSupplierBranchInfo	SUPPLIER, SUPPLIERBRANCH	
Ramificação de fornecedores	viewSupplierBranchToDelete (chama fetchSupplierBranchInfo)		SupplierBranchDelete.jsp
Ramificação de fornecedores	viewSupplierBranchToUpdate (chama fetchSupplierBranchInfo)		SupplierBranchUpdate.jsp
Contacto do fornecedor	fetchSupplierContactInfo	SUPPLIER, SUPPLIERBRANCH, SUPPLIERCONTACT	
Contacto do fornecedor	viewSupplierContactToDelete (chama fetchSupplierContactInfo)		SupplierContactDelete.jsp
Contacto do fornecedor	viewSupplierContactToUpdate (chama fetchSupplierContactInfo)		SupplierContactUpdate.jsp

**Tabela 6: Métodos para preparar beans associados a sessões com informações actuais ou para chamar métodos que preparem beans associados a sessões com informações actuais. As informações são então apresentadas na página apropriada.**

Os restantes métodos são descritos em detalhe:

- **getInventoryCreateInfo** - O método getInventoryCreateInfo obtém uma lista de UPCs da lista PRODUCT. Estas informações são apresentadas no JSP InventoryCreate.
- **getInventorySearchInfo** - O método getInventorySearchInfo obtém uma lista de UPCs válidos da tabela PRODUCT e todas as localizações de armazéns válidas da tabela INVENTORY. Estas informações são apresentadas no JSP InventorySearch.
- **getProductCreateInfo** - O método getProductCreateInfo obtém informações nos produtos que estão na tabela SUPPLIERPRODUCT mas não na tabela PRODUCT. Estas informações são apresentadas no JSP ProductCreate.
- **getProductSearchInfo** - O método getProductSearchInfo obtém uma lista dos tipos de pacote válidos para fornecer valores assumidos para a pesquisa. Estas informações são apresentadas no JSP ProductSearch.
- **fetchCostLimitMin** - O método fetchCostLimitMin calcula o limite de custo inferior a que o código de transporte pode ser aplicado e define estas informações no Bean Transporte. Este método é chamado pelo método fetchShippingInfo.
- **fetchSupplierIDs** - O método fetchSupplierIDs obtém uma lista dos IDs e nomes de fornecedor válidos para fornecer valores assumidos para a pesquisa. Estas informações são apresentadas no JSP SupplierBranchSearch.
- **preCreateSupplierBranch** - O método preCreateSupplierBranch chama o método fetchSupplierIDs para obter uma lista de fornecedores válidos. Estas informações são apresentadas no JSP SupplierBranchCreate.
- **preSearchSupplierBranch** - O método preSearchSupplierBranch chama o método fetchSupplierIDs para obter uma lista de fornecedores válidos. Estas informações são apresentadas no JSP SupplierBranchSearch.

- **fetchSupplierBranchIDs** - O método fetchSupplierBranchIDs obtém uma lista de IDs e de nomes de ramificações de fornecedores válidos para fornecer valores assumidos para a pesquisa. Estas informações são apresentadas no JSP SupplierContactSearch.
- **preCreateSupplierContact** - O método preCreateSupplierContact chama o método fetchSupplierBranchIDs para obter uma lista de ramificações de fornecedores válidos. Estas informações são apresentadas no JSP SupplierContactCreate.
- **preSearchSupplierContact** - O método preSearchSupplierContact chama os métodos fetchSupplierIDs e fetchSupplierBranchIDs para obter uma lista de fornecedores válidos e ramificações de fornecedores. Estas informações são apresentadas no JSP SupplierContactSearch.

## JavaServer Pages

As seguintes tabelas contêm os JSPs que são utilizados por EmployeeServlet. Uma vez que a maior parte das entidades da iDrink estão associadas a JSPs que executam acções semelhantes, os JSPs são categorizados pelas informações apresentadas ou solicitadas.

Entidade	Nome do JSP	Método que chama o JSP	Página que contém uma ligação ao JSP
Inventário	InventoryCreate.jsp		InventoryHome.jsp
Produto	ProductCreate.jsp		ProductHome.jsp
Transporte	ShippingCreate.jsp		ShippingHome.jsp
Fornecedor	SupplierCreate.jsp		SupplierHome.jsp
Ramificação de fornecedores	SupplierBranchCreate.jsp		SupplierHome.jsp
Contacto do fornecedor	SupplierContactCreate.jsp		SupplierHome.jsp

**Tabela 7: JSPs que pedem informações para criar um novo objecto.**

Entidade	Nome do JSP	Método que chama o JSP	Página que contém uma ligação ao JSP
Inventário	InventoryDelete.jsp	viewInventoryToDelete	
Produto	ProductDelete.jsp	viewProductToDelete	
Transporte	ShippingDelete.jsp	viewShippingToDelete	
Fornecedor	SupplierDelete.jsp	viewSupplierToDelete	
Ramificação de fornecedores	SupplierBranchDelete.jsp	viewSupplierBranchToDelete	
Contacto do fornecedor	SupplierContactDelete.jsp	viewSupplierContactToDelete	

**Tabela 8: JSPs que eliminam objectos.**

Entidade	Nome do JSP	Método que chama o JSP	Página que contém uma ligação ao JSP
Cliente	EmployeeCustomerInfo.jsp	viewCustomer	
Cliente	EmployeeCustomerOrderInfo.jsp	viewCustomerOrder	
Inventário	InventoryInfo.jsp	viewInventoryItem	
Produto	ProductInfo.jsp	viewProductItem	

Transporte	ShippingInfo.jsp	viewShippingItem	
Fornecedor	SupplierInfo.jsp	viewSupplierItem	
Ramificação de fornecedores	SupplierBranchInfo.jsp	viewSupplierBranchItem	
Contacto do fornecedor	SupplierContactInfo.jsp	viewSupplierContactItem	

**Tabela 9: JSPs que apresentam informações detalhadas sobre um objecto.**

Entidade	Nome do JSP	Método que chama o JSP	Página que contém uma ligação ao JSP
Inventário	InventoryHome.jsp		EmployeeHome.jsp
Produto	ProductHome.jsp		EmployeeHome.jsp
Transporte	ShippingHome.jsp		EmployeeHome.jsp
Fornecedor, Ramificação de fornecedores, Contacto do fornecedor	SupplierHome.jsp		EmployeeHome.jsp

**Tabela 10: JSPs que são páginas iniciais de várias entidades.**

Entidade	Nome do JSP	Método que chama o JSP	Página que contém uma ligação ao JSP
Cliente	EmployeeCustomerOrdersSearch.jsp		EmployeeHome.jsp
Inventário	InventorySearch.jsp		InventoryHome.jsp
Produto	ProductSearch.jsp		ProductHome.jsp
Transporte	ShippingSearch.jsp		ShippingHome.jsp
Fornecedor	SupplierSearch.jsp		SupplierHome.jsp
Ramificação de fornecedores	SupplierBranchSearch.jsp		SupplierHome.jsp
Contacto do fornecedor	SupplierContactSearch.jsp		SupplierHome.jsp

**Tabela 11: JSPs que permitem aos empregados pesquisar objectos.**

Entidade	Nome do JSP	Método que chama o JSP	Página que contém uma ligação ao JSP
Cliente	EmployeeCustomerList.jsp	searchCustomers	
Cliente	EmployeeCustomerOrdersList.jsp	searchCustomerOrders	
Inventário	InventorySearchResults.jsp	searchInventory	
Produto	ProductSearchResults.jsp	searchProduct	
Transporte	ShippingSearchResults.jsp	searchShipping	
Fornecedor	SupplierSearchResults.jsp	searchSupplier	
Ramificação de fornecedores	SupplierBranchSearchResults.jsp	searchSupplierBranch	
Contacto do fornecedor	SupplierContactSearchResults.jsp	searchSupplierContact	

**Tabela 12: JSPs que apresentam objectos que correspondem aos critérios especificados.**

Entidade	Nome do JSP	Método que chama o JSP	Página que contém uma ligação ao JSP
Inventário	InventoryUpdate.jsp	viewInventoryToUpdate	
Produto	ProductUpdate.jsp	viewProductToUpdate	
Transporte	ShippingUpdate.jsp	viewShippingToUpdate	
Fornecedor	SupplierUpdate.jsp	viewSupplierToUpdate	
Ramificação de fornecedores	SupplierBranchUpdate.jsp	viewSupplierBranchToUpdate	
Contacto do fornecedor	SupplierContact.jsp	viewSupplierContactToUpdate	

**Tabela 13: JSPs que pedem informações para actualizar um objecto existente.**

## SupplierServlet

O SupplierServlet da iDrink fornece a funcionalidade que os fornecedores da iDrink necessitam para gerir produtos fornecidos à iDrink. Oferece aos fornecedores a capacidade de:

- visualizar informações detalhadas acerca dos produtos que fornecem à iDrink;
- adicionar novos produtos que a iDrink pode escolher para adicionar às listagens de produtos;
- aditar informações de produtos (se a iDrink ainda não estiver a comprar o produto ao fornecedor);
- pesquisar ramificações e contactos do fornecedor;
- visualizar informações detalhadas sobre ramificações e contactos do fornecedor.

O SupplierServlet e os JSPs que chama trabalham com o UserServlet para fornecer a funcionalidade de assegurar que apenas os fornecedores podem iniciar sessão e aceder às páginas associadas ao SupplierServlet.

### Pontos de concepção

- Como foi referido na secção EmployeeServlet, muitos dos métodos e JSPs necessários entre EmployeeServlet e SupplierServlet eram semelhantes. De forma a evitar a duplicação do código, estes métodos e JSPs foram codificados apenas uma vez. Os métodos comuns foram codificados no EmployeeServlet. Consulte a secção EmployeeServlet para obter mais detalhes.
- Um fornecedor apenas pode editar produtos que a iDrink não tenha actualmente em produção. Um produto está em produção para a iDrink se o produto estiver na tabela PRODUCT. As páginas JSP reflectem isto quando verificam um campo no SupplierProductBean chamado isEditable. Este campo está definido como verdadeiro se o produto estiver actualmente em produção para a iDrink e como falso se não estiver. O seguinte código do SupplierProductListing.jsp mostra como este campo é verificado. (No código apresentado a seguir, spb é um SupplierProductBean):

```
<td>
<%
if(spb.isEditable()) {
%>
<b><a href="SupplierServlet?action=productedit&upc=<%=spb.getUPC()%>">edit</a></b>
<%
} else {
%>
<span class="noteditable">in distribution</span>
<%
}
%>
</td>
```

- Para apresentar apenas os produtos fornecidos pelo fornecedor com sessão iniciada e não apresentar os produtos fornecidos por outros fornecedores, o ID do fornecedor foi utilizado como uma marca na instrução de SQL:

```
String queryString = "select
idrink.product.upc as pupc,
idrink.supplierproduct.suppliercontactid,
idrink.supplierproduct.upc,
idrink.supplierproduct.brand,
idrink.supplierproduct.name,
idrink.supplierproduct.size,
idrink.supplierproduct.packagetype
from idrink.supplierproduct,
idrink.suppliercontact,
idrink.supplierbranch
left outer join idrink.product
on idrink.product.upc=idrink.supplierproduct.upc
where idrink.supplierproduct.suppliercontactid=idrink.suppliercontact.suppliercontactid
and idrink.suppliercontact.branchid=idrink.supplierbranch.branchid
and idrink.supplierbranch.supplierid=?";
```

### Fluxo da aplicação

A interface do fornecedor da iDrink não impões um caminho predeterminado aos fornecedores. Desta forma, um fornecedor pode facilmente passar de uma tarefa para outra em qualquer altura. Como consequência desta flexibilidade, é muito difícil ilustrar todos os caminhos possíveis que um fornecedor pode percorrer através da interface do fornecedor. A figura 10 ilustra o fluxo da aplicação básico disponível para os fornecedores da iDrink.

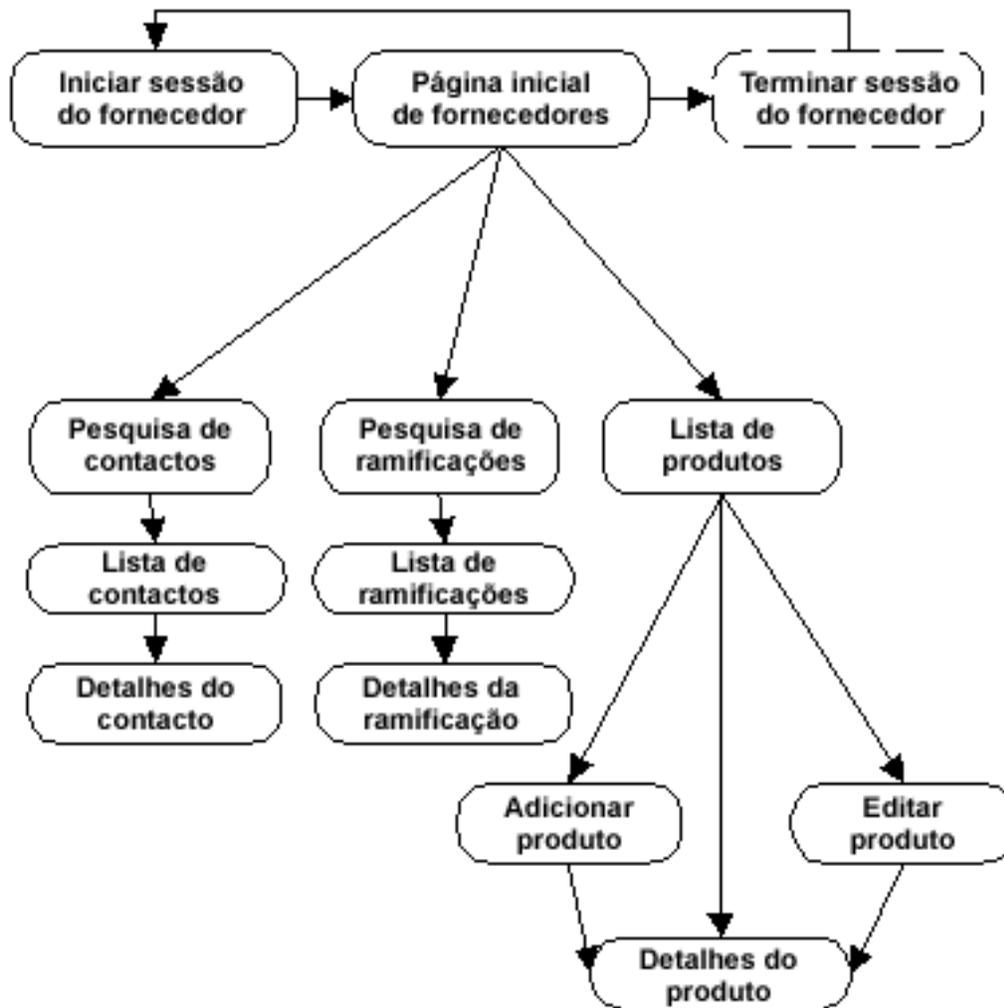


Figura 10: Fluxo da aplicação de fornecedores da iDrink

### Métodos

A seguir é apresentada uma lista de métodos que se encontram no SupplierServlet. O método doPost recebe todos os pedidos chegados do cliente e chama o método apropriado a partir desta lista de métodos.

- **addSupplierProduct** - O método addSupplierProduct insere um novo produto na tabela SUPPLIERPRODUCT com base nos dados do formulário/pedido.
- **displaySupplierProductDetails** - O método displaySupplierProductDetails chama o método fetchSupplierProductDetails para preparar o SupplierProductBean associado à sessão com informações actuais. Estas informações são apresentadas no SupplierProductInfo.jsp.
- **editSupplierProduct** - O método editSupplierProduct actualiza as informações dos produtos do fornecedor com valores fornecidos pelo mesmo.
- **fetchSupplierProductDetails** - O método fetchSupplierProductDetails prepara o SupplierProductBean associado à sessão com as informações necessárias para apresentar o produto do fornecedor actual (apenas no caso de as alterações terem sido efectuadas no intervalo entre as informações serem apresentadas e a necessidade de actualização das mesmas).
- **loadSupplierContacts** - O método loadSupplierContacts obtém uma lista de contactos do fornecedor válidos para o fornecedor actual.

- **prepareForSupplierProductAdd** - O método `prepareForSupplierProductAdd` chama o método `loadSupplierContacts` para obter uma lista de contactos do fornecedor actual. Estas informações são apresentadas no `SupplierProductAdd.jsp`.
- **prepareForSupplierProductEdit** - O método `prepareForSupplierProductEdit` chama o método `fetchSupplierProductDetails` para preparar o `SupplierProductBean` associado à sessão com informações actuais. Estas informações são apresentadas no `SupplierProductEdit.jsp`.
- **searchSupplierProducts** - O método `searchSupplierProducts` obtém produtos fornecidos pelo fornecedor com sessão iniciada. Estas informações são apresentadas no `SupplierProductListing.jsp`.

## JavaServer Pages

A seguir é apresentada uma lista de JSPs utilizados por fornecedores (os JSPs marcados com um \* são também utilizados no fluxo da aplicação de empregados):

- **\*SupplierBranchInfo** - O JSP `SupplierBranchInfo` apresenta informações detalhadas sobre uma ramificação de fornecedores específica.
- **\*SupplierBranchSearch** - O JSP `SupplierBranchSearch` fornece campos de entrada de dados que permitem a um fornecedor introduzir e submeter um conjunto de critérios de pesquisa de ramificações.
- **\*SupplierBranchSearchResults** - O JSP `SupplierBranchSearchResults` apresenta todas as ramificações de fornecedores que correspondem aos critérios de pesquisa especificados.
- **\*SupplierContactInfo** - O JSP `SupplierContactInfo` apresenta informações detalhadas sobre um contacto do fornecedor específico.
- **\*SupplierContactSearch** - O JSP `SupplierContactSearch` fornece campos de entrada de dados que permitem a um fornecedor introduzir e apresentar um conjunto de critérios de pesquisa de contactos.
- **\*SupplierContactSearchResults** - O JSP `SupplierContactSearchResults` apresenta todos os contactos do fornecedor que correspondem aos critérios de pesquisa especificados.
- **SupplierHome** - O JSP `SupplierHome` é a página inicial do fluxo da aplicação de fornecedores.
- **SupplierProductAdd** - O JSP `SupplierProductAdd` fornece campos de entrada de dados que permitem a um fornecedor criar um novo produto do fornecedor.
- **SupplierProductEdit** - O JSP `SupplierProductEdit` fornece campos de entrada de dados que permitem a um fornecedor editar um produto do fornecedor existente.
- **SupplierProductInfo** - O JSP `SupplierProductInfo` apresenta informações detalhadas sobre um produto do fornecedor específico.
- **SupplierProductListing** - O JSP `SupplierProductListing` apresenta todos os produtos que o fornecedor fornece.

---

## Instalação da aplicação para empresas

A instalação de uma aplicação para empresas numa instância do IBM<sup>(R)</sup> WebSphere<sup>(R)</sup> Application Server é tratada pelos assistentes que se encontram no WebSphere Studio Application Developer (WSAD), na IBM WebSphere Administrative Console (Consola Administrativa do WebSphere da IBM) e na IBM Web Administration for iSeries<sup>(TM)</sup> Console (Consola de Administração da Web para iSeries da IBM). Para implementar uma aplicação, esta deve primeiro ser exportada para um ficheiro EAR (Enterprise ARchive) na Plataforma Java<sup>(TM)</sup> 2, Enterprise Edition (J2EE). O ficheiro EAR contém todos os ficheiros de classes e informações adicionais necessárias para instalar a aplicação num sistema de produção e pode ser criado por um assistente no WSAD. Depois de o ficheiro EAR ter sido criado, o programador pode utilizar a Consola Administrativa do WebSphere ou a Consola de Administração da Web para iSeries da IBM para implementar o ficheiro EAR no sistema de produção.

### Exportar a aplicação para um ficheiro EAR

Quando uma aplicação para empresas tiver sido compilada sem erros, um programador exportará a aplicação para um ficheiro EAR, o qual será colocado no directório **installableApps** (`\QIBM\UserData\WebASE51\ASE\Nomeinstância\installableApps`) da instância WebSphere Application

Server. Este processo está documentado em *Section 13.8.1 - Exporting an enterprise application to a file location* no Redbook *IBM WebSphere Application Server - Express V5.0.2 Developer Handbook, SG24-6555*.

**Implementar o ficheiro EAR file numa instância do IBM WebSphere Application Server** Quando o ficheiro EAR tiver sido criado com êxito pelo assistente do WSAD, o ficheiro EAR pode ser implementado num sistema de produção. Um programador pode optar por utilizar a Consola Administrativa do WebSphere ou a Consola de Administração da Web para iSeries da IBM para executar esta tarefa. O processo de implementação do ficheiro EAR através da Consola Administrativa do WebSphere está documentado em *Section 6.4 - Installing applications* no Redbook *WebSphere Application Server - Express V5.0.2 Administrator Handbook, SG24-6976*. Para implementar o ficheiro EAR através da Consola de Administração da Web para iSeries da IBM, o ficheiro EAR já tem de estar no sistema de destino. Para implementar o ficheiro EAR, siga estes passos:

- In a navegador da Web, avance para a Consola de Administração da Web para iSeries da IBM no sistema de destino. O URL é `http://meusistema:2001/HTTPAdmin`, em que “meusistema” é o nome do sistema de destino. Introduza um ID de utilizador e uma palavra-passe para o sistema de destino quando lhe forem pedidas essas informações.
- Seleccione o separador **Manage** (Gerir) na parte superior da página e, em seguida, seleccione o separador **Application Servers** (Servidores da aplicação) abaixo.
- Certifique-se que a instância do servidor WebSphere Application Server está seleccionada no menu **Instance/Server** (Instância/Servidor). Se a instância do servidor estiver parada, faça clique no botão iniciar de cor verde à esquerda do menu para iniciar a instância do servidor. Quando a instância do servidor estiver em **execução**, avance para o passo seguinte para começar a instalação da aplicação.
- Na estrutura à esquerda, clique na ligação **Install New Application** (Instalar nova aplicação).
- No ecrã **Specify Application Location** (Especificar localização da aplicação), faça clique no botão **Browse** (Navegar). Na janela do navegador, seleccione o ficheiro EAR e faça clique no botão **OK**. Faça clique no botão **Next** (Seguinte) no ecrã **Specify Application Location** (Especificar localização da aplicação).
- No ecrã **Provide Options to Perform Install** (Fornecer opções para executar a instalação), introduza um nome da aplicação no campo de texto **Application Name** (Nome da aplicação) e marque o selector de confirmação **Pre-compile JSPs** (Pré-compilar JSPs) se for necessário pré-compilar os JSPs. Faça clique no botão **Next** (Seguinte).
- No ecrã **Map Virtual Hosts for Web Modules** (Definir correspondência de sistemas centrais virtuais para módulos da Web), seleccione **default\_host** no menu **Virtual host** (Sistema central virtual). Faça clique no botão **Next** (Seguinte).
- No ecrã **Summary** (Resumo), verifique se as informações estão correctas e faça clique no botão **Finish** (Terminar). A instalação da aplicação pode demorar alguns minutos.

---

## Observações do WebSphere

- Antes do WebSphere Application Server - Express for iSeries<sup>(TM)</sup>, versão 5.1, os JSPs conseguiam importar classes do pacote assumido. No entanto, com o WebSphere Application Server - Express for iSeries, versão 5.1, os JSPs deixaram de poder importar classes do pacote assumido. Inicialmente, os servlets e JavaBeans<sup>(TM)</sup> da iDrink estavam localizados no pacote assumido. Para os mover para um pacote não assumido, siga estes passos no WebSphere Studio Application Developer:
  - Crie um novo pacote e mova os ficheiros fonte de java para o novo pacote (WebSphere Studio Application Developer adicionará a instrução do “pacote” apropriada a todos os ficheiros fonte quando estes são movidos)
  - Quando um código criar uma instância de bean, inclua o nome do pacote com o nome do bean, por exemplo  

```
Beans.instantiate(getClass().getClassLoader(), "packageName.beanName");
```
  - Quando um JSP importar um bean, inclua o nome do pacote com o nome do bean, por exemplo  

```
<%@ page import="packageName.beanName" %>
```

- Atualize a lista de servlets no ficheiro web.xml para que este aponte para servlets que incluem o nome do pacote. Apesar de o WebSphere Studio Application Developer adicionar a instrução do pacote apropriado a cada um dos ficheiros fonte quando estes são movidos, não actualiza o ficheiro web.xml para apontar para os servlets no novo pacote. Se utilizar o editor GUI, remova os servlets da lista e adicione-os novamente. O campo "Servlet class" (Campo do servlet) deve conter packageName.servletName.

### **Referências**

- WebSphere Application Server - Express V5.0 for iSeries, IBM<sup>(R)</sup> Redpaper REDP-3624-00
- WebSphere J2EE Application Development for the IBM eServer<sup>(TM)</sup> iSeries Server, IBM Redbook SG24-6559-00
- WebSphere Development Studio Client for iSeries V5.0, IBM Redbook SG24-6961-00
- WebSphere Studio Application Developer Programming Guide, IBM Redbook SG24-6585-00

---

## Capítulo 3. Ambiente do Lotus Domino

A empresa iDrink pretendia fornecer um sítio da Web no qual os empregados pudessem procurar e colocar anúncios classificados. Este sítio da Web, Cola Connections, foi criado com o Lotus Domino versão 6.5 no eServer<sup>(TM)</sup> i5.

Com o Lotus Domino, a empresa iDrink conseguia criar rapidamente um sítio da Web dinâmico de anúncios classificados. Além disso, o Lotus Domino permitiu que a empresa iDrink beneficiasse do processamento do fluxo de trabalho e forneceu interoperacionalidade com muitas outras plataformas.

A aplicação Cola Connections utiliza vários formatos e vistas para criar e manter informações publicitárias. As informações publicitárias são anúncios classificados listados para venda colocados pelos empregados. Para gerir com segurança os dados do empregado, é utilizada uma base de dados de assistência de directório do Lotus Domino com referência a um directório do i5/OS<sup>(TM)</sup> Lightweight Directory Access Protocol (LDAP).

A empresa iDrink fornece uma interface Web utilizando conjuntos de estruturas do Lotus Domino. Através do sítio da Web Cola Connections, os empregados da iDrink podem executar as seguintes tarefas:

- Visualizar anúncios classificados
- Criar anúncios classificados
- Editar anúncios classificados existentes com uma opção para assinalar o anúncio classificado como vendido

---

### Configurar o Lotus Domino para utilizar o LDAP do OS/400

Uma vez que a segurança era um requisito do sítio da Web da Cola Connections, o Lotus Domino foi configurado para utilizar os serviços LDAP do i5/OS<sup>(TM)</sup> para autenticação do utilizador. Como valor assumido, o Domino utiliza a própria capacidade LDAP. Para o Domino utilizar o serviço LDAP do i5/OS em vez da própria capacidade LDAP, é necessário criar uma base de dados de assistência de directório.

Para obter instruções detalhadas sobre como configurar uma base de dados de assistência de directório, consulte a secção 4.4.1 *Creating a Directory Assistance database* no Redbook intitulado *Integrating Lotus Domino 6 and WebSphere<sup>(R)</sup> Express V5 on the IBM<sup>(R)</sup> eServer<sup>(TM)</sup> iSeries<sup>(TM)</sup> Server (SG24-6998-00)*.

Estas sugestões são úteis para configurar a base de dados de assistência de directório da Cola Connections ao seguir as instruções contidas no Redbook:

- No passo 8, ao introduzir os campos no separador **Geral**, pode especificar qualquer nome para o domínio excepto para o domínio utilizado pelo servidor Domino. Por exemplo, o servidor Domino da iDrink foi configurado no domínio iDrink. Deste modo, no separador **Basics** (Básico), não foi possível especificar iDrink como o nome do domínio.
- No passo 10, ao introduzir os campos no separador **LDAP**, com base na configuração LDAP para a iDrink, no campo "Type of search filter to use" (Tipo de filtro de pesquisa a utilizar) especifique custom (personalizado) e na secção "Customized filters" (Filtros personalizados) adicione o seguinte ao campo "Authorization Filter" (Filtro de autorização):

```
(|(&(objectclass=groupofuniqueNames)(UniqueMember=%*))(&(objectclass=groupofNames)(objectclass=AccessGroup)(member=%*)))
```

Este filtro de pesquisa permite ao Domino pesquisar grupos LDAP. A Lista de controlo de acesso (ACL, Access Control List) na base de dados Domino é, então, definida de modo a que apenas o empregado possa aceder à mesma. Uma vez que todos os empregados pertencem a este grupo, podem agora aceder à Aplicação Cola Connections.

Continue a configurar o servidor Domino seguindo as instruções na secção "4.4.2 Configurar o Domino para utilizar o LDAP do OS/400" até ao final do capítulo.

Para configurar a ACL na base de dados do Domino, siga as instruções na secção "5.2.3 Actualizar ACLs do Domino para adicionar utilizadores registados no LDAP do OS/400". Por exemplo, a base de dados Cola Connections utilizou a seguinte ACL do grupo de pessoas a partir do esquema LDAP da iDrink: `cn=employees/o=drink/dc=domainName/dc=domainSuffix`. As barras têm de ser introduzidas como se mostra acima para que o Domino consiga encontrar o grupo no esquema LDAP.

---

## Detalhes da aplicação

No sítio da Web da Cola Connections, um empregado da iDrink pode executar diversas tarefas. Um empregado da iDrink pode criar um anúncio premindo o botão **New Ad** (Novo anúncio). Depois de criado, este novo documento pode então ser actualizado, marcado como vendido ou apresentado.

---

## Pontos de concepção de aplicações

Para conceber a aplicação Lotus<sup>(R)</sup> Domino<sup>(R)</sup>, a empresa iDrink teve de escolher entre utilizar conjuntos de estruturas ou navegadores para a interface da Web. A seguir são apresentadas as diferenças entre conjuntos de estruturas e navegadores:

- Os conjuntos de estruturas podem conter formulários, pastas, páginas, vistas, navegadores, ou conjuntos de estruturas. A estrutura pode também conter uma página da Web e estar associada a um endereço da Web específico. As ligações e as relações entre estruturas podem ser criadas através de conjuntos de estruturas.
- Os navegadores fornecem uma apresentação gráfica de pastas, vistas e elementos de concepção para que a maior parte dos utilizadores encontrem as informações mais facilmente.

Uma vez que a empresa iDrink pretendia oferecer um sistema consistente em todo o sítio da Web onde se pudessem visualizar diferentes formulários e vistas e como as estruturas podem conter fichas, pastas, páginas, documentos e vistas, o conjunto de estruturas constituía a melhor opção.

Outra tarefa enfrentada pela empresa iDrink era encontrar e implementar a melhor forma de proteger os respectivos dados. Os requisitos de segurança da Cola Connections incluíam:

- uma arquitectura robusta que oferecesse fiabilidade, velocidade de implementação e uma facilidade de administração;
- uma solução que integrasse facilmente as aplicações existentes;
- uma solução que pudesse ser integrada com outros pacotes de software.

Estes requisitos conduziram a empresa iDrink a implementar uma base de dados de assistência de directório no Domino Server com referência ao directório LDAP do i5/OS<sup>(TM)</sup>.

Uma vez que a empresa iDrink tem vários empregados com competência no desenvolvimento de Java<sup>(TM)</sup>, os agentes de segundo plano da aplicação foram escritos em Java. Isto acabou por ser valioso para a utilização das competências e dos recursos existentes.

A figura 11 apresenta o fluxo da aplicação Cola Connections.

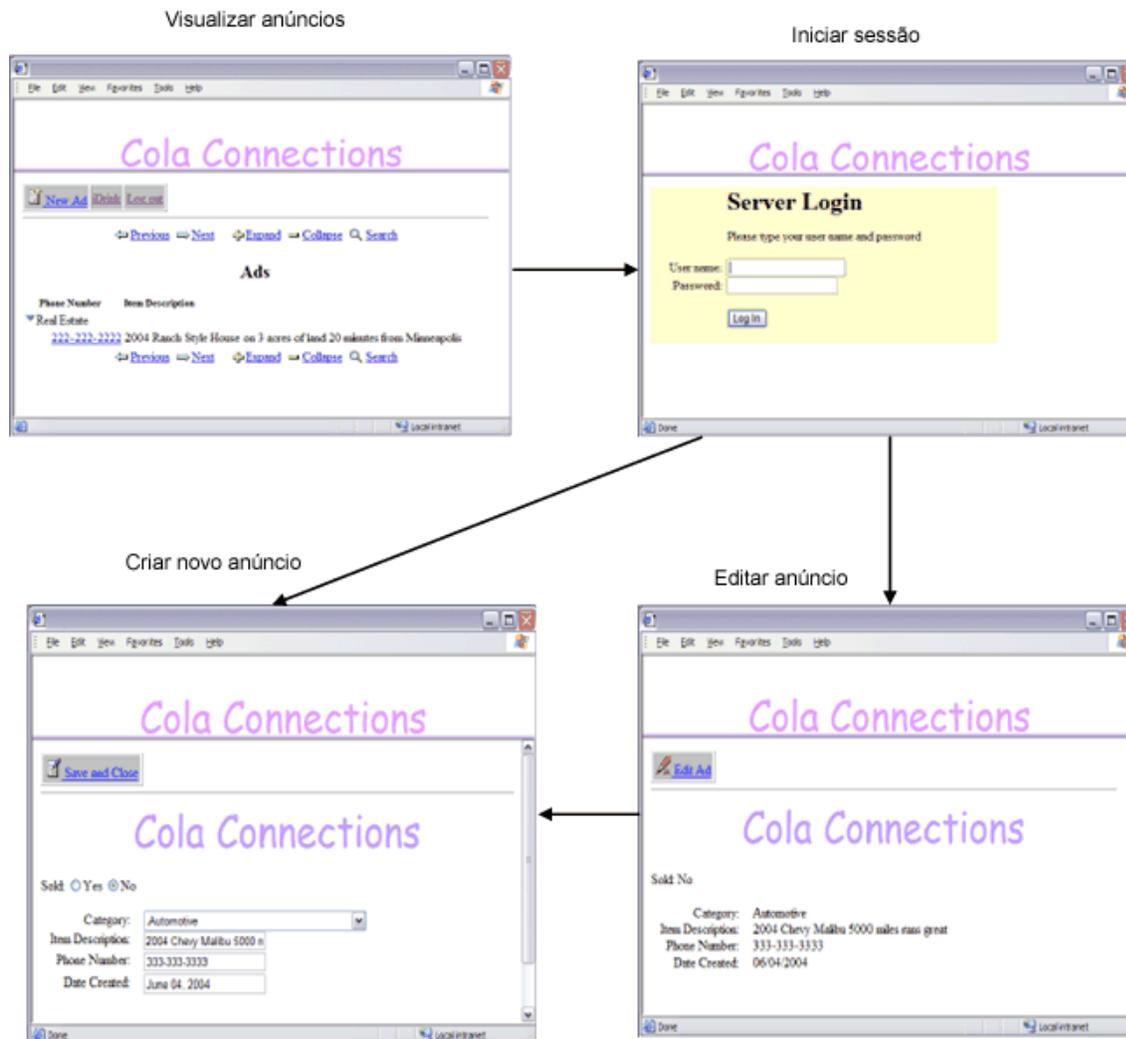


Figura 11: Fluxo da aplicação Cola Connections

## Instalação da aplicação

O Lotus<sup>(R)</sup> Domino<sup>(R)</sup> Designer<sup>(R)</sup> foi utilizado como ferramenta de desenvolvimento para o trabalho de implementação do Lotus Domino. O Lotus Domino Designer é uma aplicação cliente do Lotus Notes<sup>(R)</sup> que pode ser utilizada para criar e modificar rapidamente as aplicações Lotus Domino. Fornece os blocos de construção de aplicações para todos os elementos da base de dados, incluindo formulários, vistas e agentes. Os formulários foram utilizados na iDrink para criar novos documentos numa base de dados e apresentar os documentos actuais. As vistas fornecem uma forma flexível e intuitiva de organizar documentos. Os utilizadores podem facilmente criar, ordenar, visualizar e editar documentos.

## Formulários e vistas do Lotus Domino

Os empregados da empresa iDrink criaram o formulário Informações dos anúncios e as vistas Anúncios e Todos os anúncios para a aplicação Lotus Domino.

A vista Informações dos anúncios contém a lista de todos os anúncios disponíveis que foram criados pelos empregados da iDrink. O formulário Informações dos anúncios é utilizado para criar um anúncio novo ou para apresentar um já existente.

As vistas Anúncios e Todos os anúncios apresentam os diversos anúncios e estão ordenadas por categorias como, por exemplo, automóvel, informática/escritório em casa e imobiliária.

A tabela 13 apresenta os campos do formulário Informações dos anúncios.

Nome	Nome do campo	Tipo	Descrição
Categoria	Category	Lista de diálogo	A categoria do Anúncio. As categorias possíveis são: Automóvel, Automóvel - Diversos, Barcos/Autocaravanas/Motos de neve & Outras viaturas, Vestuário, Informática/Escritório em casa, Lar, Crianças, Diversos, Motocicleta, Música, Fotografia/Som/TV, Artigos de Desporto, Imobiliária, Comércio, Procura, Ofertas
Descrição do artigo	ItemDescription	Texto	A descrição do artigo para o anúncio. Limitada a 100 caracteres
Número de telefone	PhoneNumber	Texto	O número de telefone do empregado a vender o artigo
Data da criação	DateCreate	Data/Hora	Campo gerado automaticamente que indica a data em que o Anúncio foi criado
Vendido	Sold	Selector de opção	Indica se o artigo foi vendido. Os valores possíveis são "Sim" ou "Não"

**Tabela 13: Campos no formulário Informações dos anúncios.**

## Agentes do Lotus Domino

Os agentes do Lotus Domino são elementos de concepção adicionados a uma base de dados do Lotus Domino para automatizar tarefas. Os agentes podem ser iniciados através de uma execução ou acção do utilizador numa base programada. Normalmente, os agentes são utilizados para actualizar ou criar documentos ou para aceder a bases de dados. Os agentes do Lotus Domino podem ser escritos em Java<sup>(TM)</sup>, LotusScript ou Formula Language.

A criação de agentes requer o Lotus Domino Designer<sup>(R)</sup>. As decisões que devem ser tomadas durante a criação de um agente incluem a altura em que este deve ser executado, a linguagem em que deve ser escrito e em que documentos deve ser executado. Depois de o agente ter sido escrito e compilado é automaticamente programado para ser executado na altura especificada. As capacidades de depuração incorporadas do Lotus Domino Designer são úteis para escrever os agentes.

A aplicação Lotus Domino Cola Connections contém um agente, Eliminar anúncios. Este Agente Java é executado numa base programada. Eliminará os documentos de anúncios que tenham estado de venda de sim ou que tenham mais do que um mês.

A seguir é apresentado um extracto do código do agente Eliminar anúncios:

```
private void deleteAd(AgentContext agentContext, Session session) {

    Vector adDateVector;
    DateTime adDateTime = null;
    Date adDate = null;
    Date currentDate = null;
    int timeBetween = 0;
    float timePassed = 0;
```

```

try {
    Database db = agentContext.getCurrentDatabase();
    DateTime currentDateTime = session.createDateTime("Today"); // get the current date
    currentDateTime.setNow(); // get all documents where the Sold Status is Yes from All Ads
    DocumentCollection adInformationDC = db.search("SELECT ((Form = \"Ad Information\"))");
    Document adDoc = adInformationDC.getFirstDocument(); // get first ad document
    while (adDoc != null) // while there are ad documents
        String adStatus = adDoc.getItemValueString("Sold"); // obtain the value of the Sold variable
        adDateVector = adDoc.getItemValue("DateCreate"); // obtain value of create date for ad
        adDateTime = (DateTime)adDateVector.elementAt(0); // obtain value of create date for ad
        timeBetween = currentDateTime.timeDifference( adDateTime); // returns difference in seconds
        timePassed = timeBetween/2629744; //divide by seconds in a month
        Document adDoc1 = adInformationDC.getNextDocument(); // get the next ad document
        if ((adStatus.equals("Yes")) || (timePassed > 1)) // if item is Sold, remove the document
            adDoc.remove(true);
        adDoc = adDoc1; // get the next ad to process
    } // end while adDoc is not null }
catch(Exception e) {
    e.printStackTrace();
}} // end delete ads

```

---

## Observações do Lotus Domino

A seguir é apresentada uma lista das principais observações efectuadas durante a criação do cenário Cola Connections:

- Depois da instalação inicial, não era pedido aos empregados da iDrink que iniciassem sessão ao criar um novo anúncio. Por vezes, o pedido de início de sessão só aparece quando o botão Save and Close (Guardar e fechar) era premido. Para pedir a um empregado para iniciar sessão quando premir o botão New Ad (Novo anúncio), foi definido o seguinte comando do idioma das fórmulas na acção New Ad (Novo anúncio):

```
@URLOpen(@WebDbName + "/Ad+Information?OpenForm&login")
```

- A empresa iDrink pretendeu implementar uma forma de os empregados concluírem sessão da aplicação Cola Connections. Para este efeito, foi implementada um componente de conclusão de sessão. No entanto, quando era premido o botão de conclusão de sessão, os empregados não estavam a ser redirigidos para a página inicial Cola Connections. Para activar o redireccionamento para a página inicial, foi definido o seguinte comando do idioma das fórmulas na acção Log out (Conclusão de sessão):

```
@SetTargetFrame("_top");
@URLOpen("http://nenotes:2500/ColaConn.nsf?logout&RedirectTo=http://NotesServerName:2500/ColaConn.nsf/Cola%20Connections?OpenFrameSet")
```

## Referências

- Sítio da Web IBM<sup>(R)</sup> Lotus Domino for iSeries<sup>(TM)</sup> - OS/400<sup>(R)</sup>  
<http://www.ibm.com/servers/eserver/iserries/domino>
- IBM Lotus Domino for iSeries (PartnerWorld para programadores)  
<http://www.ibm.com/servers/eserver/iserries/developer>
- Sítio da Web da Lotus  
<http://www.lotus.com>





**IBM**