

# Tecnologias de Distribuição e Integração (TDIN)

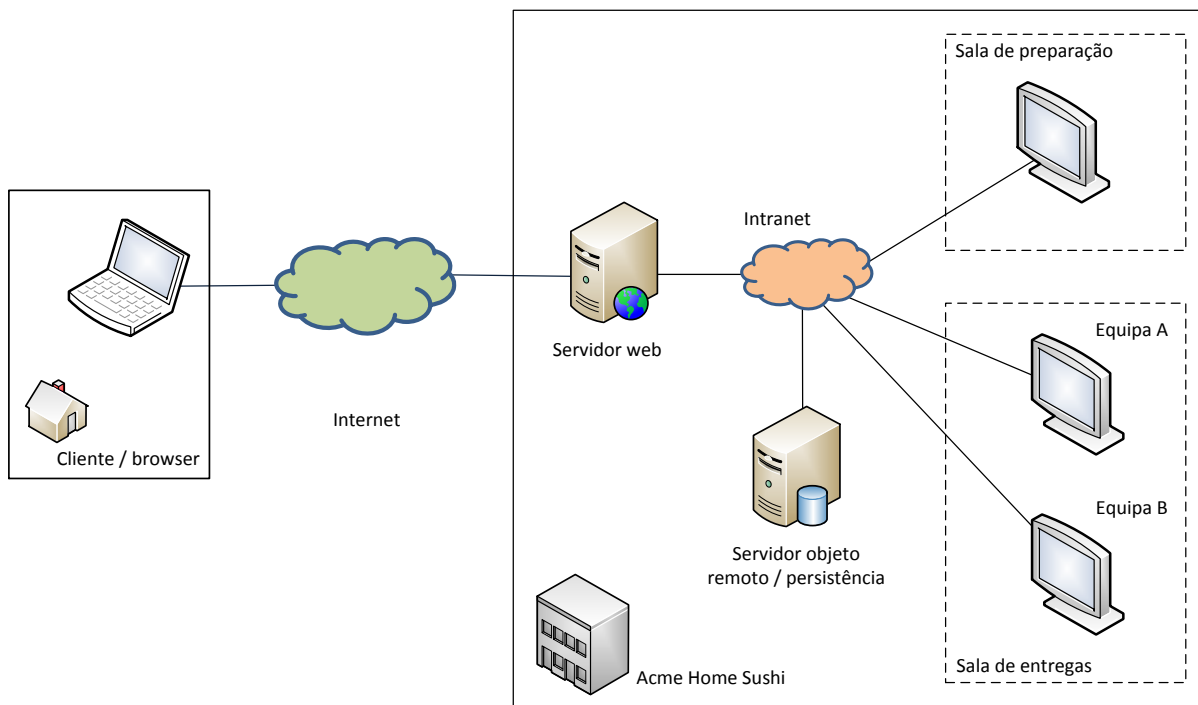
Trabalho # 1  
.NET Remoting  
2012-13

## Cenário

Uma empresa de entrega ao domicílio de sushi (Acme Home Sushi) pretende automatizar o seu sistema de encomendas, preparação e entrega.

Os clientes fazem os seus pedidos através de um sítio web. Como a empresa ainda não é muito conhecida e o número de encomendas ainda não é elevado, pretende-se que quando qualquer uma seja completada no sítio web, apareça de imediato numa lista do único terminal existente no local de preparação. Qualquer mestre preparador que esteja livre pode então tomar conta da encomenda, prepará-la, e quando estiver pronta sinalizá-la dessa forma no terminal. Quando uma encomenda é sinalizada como pronta deverá aparecer automaticamente em todos os terminais da sala de entregas. Cada um destes terminais está associado a uma equipa de entrega que a pode escolher para entrega. Quando uma equipa fizer isso, deixa de aparecer nos outros terminais. Após a entrega a equipa deve indicar isso no terminal, dando por concluída a encomenda.

Cada encomenda pode assim estar num de cinco estados: 'encomendada', 'em preparação', 'pronta para entrega', 'em entrega' e 'concluída'. Um objeto remoto num servidor deverá tomar nota de todas as encomendas e seus estados. Quando a encomenda fica concluída deve deixar de aparecer nos terminais.



## Servidor Web

O servidor web contém uma aplicação que é a interface com o utilizador. Este, para colocar uma encomenda deve fornecer os seus dados pessoais (nome, morada, cartão de crédito) e poder escolher os tipos de sushi e quantidade que vai encomendar. Antes de confirmar a encomenda deve apresentar-se o preço. Pode também visualizar as suas encomendas ativas e o seu estado.

### **Servidor do objeto remoto**

Num objeto remoto mantém-se uma lista de encomendas ativas. Esta lista deve ser persistida (por exemplo num ficheiro), quando se desliga o servidor, e recuperada quando se liga. O objeto remoto deve implementar métodos úteis aos seus clientes e também eventos, subscritos remotamente, e que devem ‘disparar’ quando algo relevante para algum cliente acontece.

### **Sala de preparação**

Contém um terminal executando uma aplicação windows e duas listas com as encomendas que chegam e ‘em preparação’. Deverá haver um mecanismo simples de interação para sinalizar uma encomenda e passá-la ao estado ‘em preparação’. O mesmo mecanismo deve passá-la ao estado de ‘pronta para entrega’. Estas encomendas devem desaparecer do terminal. Quando uma encomenda passa para ‘em preparação’ deve ser paga. Isto deve ser simulado no servidor anterior acrescentando uma linha num ficheiro de texto com a encomenda (seu identificador numérico), data e hora, nome do cliente e número do cartão de crédito.

### **Sala de entrega**

Na sala de entrega há um terminal por cada equipa de entregas. Quando se executa a aplicação (que é a mesma para todos os terminais), esta deve identificar a equipa. Esta aplicação é semelhante à anterior. Quando uma encomenda fica pronta, deve aparecer em todos os terminais ligados. Assim que uma equipa a aceite para entrega passa para a segunda lista (‘em entrega’) dessa equipa e desaparece dos outros terminais. Quando a entrega for executada, isso deve também ser sinalizado (a encomenda passa a ‘concluída’) e desaparece da lista.

### **Implementação**

Deverá implementar um demonstrador deste projeto usando .NET Remoting e tentar usar as suas boas práticas. As aplicações (web com ASP.NET e windows) devem ser muito simples de usar e interagir.

### **Relatório**

Deverá ser entregue um pequeno relatório descrevendo a arquitetura usada (aplicações, módulos e sua interação), testes, e a representação gráfica (capturas) dos principais fluxos de utilização.

Deverá conter também todas as instruções para construir e usar todo o demonstrador.