



```

mysql> DELIMITER ;
mysql>
mysql> CALL sp_calc2('Augusto Manzano');
+-----+
| Mensagem |
+-----+
| Alo, Mundo! |
+-----+
1 row in set (0.00 sec)

+-----+
| None |
+-----+
| Augusto Manzano |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER $
mysql> CREATE PROCEDURE sp_calc(IN A INTEGER, IN B INTEGER, OUT S INTEGER)
-> BEGIN
-> SET S = A + B;
-> END$
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql>
mysql> CALL sp_calc(5,3,@saida);
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @saida;
+-----+
| @saida |
+-----+
| 8 |
+-----+
1 row in set (0.00 sec)

mysql>

```

Figura 9.7 - Criação e execução da stored procedure “sp\_calc()”.

A partir do momento em que se tem uma noção básica do funcionamento estrutural de uma sub-rotina, pode-se aprender a usar tomada de decisão. A próxima rotina recebe dois valores reais com os parâmetros e apresenta o resultado da adição dos valores caso a soma seja maior ou igual a 10. Não sendo a condição satisfeita, a rotina não apresenta nada.

Considere o código da seguinte sub-rotina:

```

DELIMITER $
CREATE PROCEDURE sp_decisao1(IN A FLOAT, IN B FLOAT)
BEGIN
    DECLARE X FLOAT;
    SET X = A + B;
    IF (X >= 10) THEN
        SELECT X;
    END IF;
END$
DELIMITER ;

CALL sp_decisao1(5.5,4.5);
CALL sp_decisao1(5.5,6.5);
CALL sp_decisao1(4.5,4.5);

```

A Figura 9.8 apresenta o resultado da sub-rotina **sp\_decisao1()**. Observe que as duas primeiras chamadas com o comando **CALL** geram uma resposta de saída, já a terceira chamada nada apresenta.

O exemplo de sub-rotina seguinte apresenta o código de uma *procedure* que recebe a entrada de um parâmetro numérico inteiro e exibe uma mensagem informando se o número é ou não divisível por 3. Considere o código da seguinte sub-rotina:

```

DELIMITER $
CREATE PROCEDURE sp_decisao2(IN Valor INTEGER)
BEGIN
    DECLARE Resto INTEGER;
    SET Resto = Valor % 3;

```