

Questão 1:

Solução em PHP:

```
<?php
class ReviewUrl
{
    public $company_url = strtolower('test.trustedcompany.com');
    public $page = 0;
    public $public_key = 'd43d6dbe8a04eb719fe56b409543f174';
    public $private_key = 'a68d2d878761ec390a0d91a70cc7a056';

    public function GerarUrl($company_url, $page, $public_key, $private_key )
    {
        $key = hash_hmac('md5', $public_key . $company_url, $private_key);

        $url = 'https://trustedcompany.com/api/v1/company/reviews/' .
        $company_url . '?' . http_build_query(array('key' => $public_key . '-' . $key,
            'page' => $page));
        return $url;
    }
}
?>
```

Solução em C#

```
using System.Text;
using System.Security.Cryptography;

namespace Test
{
    public class GetReviews
    {
        string company_url;
        int page;
        string public_key;
        string private_key;

        public GetReviews()
        {
            company_url = "test.trustedcompany.com";
            page = 0;
            public_key = "d43d6dbe8a04eb719fe56b409543f174";
            private_key = "a68d2d878761ec390a0d91a70cc7a056";
        }
    }
}
```

```

    }

    public GetReviews(string company_url, int page, string public_key, string private_key)
    {
        this.company_url = company_url;
        this.page = page;
        this.public_key = public_key;
        this.private_key = private_key;
    }

    private string GerarHash()
    {
        var data = Encoding.UTF8.GetBytes(public_key + company_url);
        var key = Encoding.UTF8.GetBytes(private_key);
        var hmac = new HMACMD5(key);
        var hashbytes = hmac.ComputeHash(data);

        return System.BitConverter.ToString(hashbytes).Replace("-", "").ToLower();
    }

    public string GerarUrl()
    {
        string hash = GerarHash();

        string url = "https://trustedcompany.com/api/v1/company/reviews/" + company_url +
        "?key=" + public_key + "-" + hash + "&page=" + page;

        return url;
    }
}

```

Questão 2

Solução em PHP:

```
<?php
```

```

class UniqueLink
{
    public $company_id = 564869;
    public $company_secret = 'a907f5ed930d7d6a6778e60f7fb7acc78f5dc9e2';
    public $customer_name = 'Fabio';
}

```

```

        public $customer_email = 'fcs1983@gmail.com';
        public $order_id = 'xxxx';

        public function GerarUrl ($company_id, $company_secret, $customer_email,
        $order_id )
        {
            $hash = sha1 ($company_id . $company_secret . $customer_email .
        $order_id);
            $url = 'http://trustedcompany.com/review-invite/' . $company_id . '/ul/' .
        $hash . '?a=' . base64_encode($customer_email) . '&b=' . $customer_name . '&c=' .
        $order_id;
            return urlencode ($url);
        }
    }

?>

```

Solução em C#:

```

using System;
using System.Text;
using System.Security.Cryptography;
using System.Web;

namespace Test
{
    public class UniqueLink
    {
        int company_id;
        string company_secret;
        string customer_email;
        string customer_name;
        string order_id;

        public UniqueLink()
        {
            company_id = 564869;
            company_secret = "a907f5ed930d7d6a6778e60f7fb7acc78f5dc9e2";
            customer_email = "fcs1983@gmail.com";
            customer_name = "Fabio";
            order_id = "xxxx";
        }

        public UniqueLink(int company_id, string company_secret, string customer_email, string
        customer_name, string order_id)
    }
}

```

```

{
    this.company_id = company_id;
    this.company_secret = company_secret;
    this.customer_email = customer_email;
    this.customer_name = customer_name;
    this.order_id = order_id;
}

private string GerarHash()
{
    string encode = company_id + company_secret + customer_email + order_id;
    ASCIIEncoding AE = new ASCIIEncoding();
    byte[] HashValue, MessageBytes = AE.GetBytes(encode);
    SHA1Managed SHhash = new SHA1Managed();
    string strHex = "";
    HashValue = SHhash.ComputeHash(MessageBytes);
    strHex = BitConverter.ToString(HashValue);
    return strHex;
}

public string GerarUrl()
{
    string hash = GerarHash();

    ASCIIEncoding AE = new ASCIIEncoding();
    byte[] email_bytes = AE.GetBytes(customer_email);
    string email_base64 = Convert.ToBase64String(email_bytes);

    string url = "http://trustedcompany.com/review-invite/" + company_id + "/ul/" + hash +
"?a=" + email_base64 + "&b=" + customer_name + "&c=" + order_id;

    return HttpUtility.UrlPathEncode(url);
}
}

```

Questão 3

Select

A.CustomerName 'Customer Name',
A.PostalCode 'Postal Code',
C.EmployeeName 'Driver Name'

From Orders B

Inner Join

Customers A On B.CustomerID = A.CustomerID

Inner Join

Employee C On B.EmployeeID = C.EmployeeID;
where c.role = 'Driver';